

\$2.50

In Canada \$3.00

Recreational COMPUTING

FOR THE IMAGINATIVE SMALL COMPUTER USER!

VOL. 9 NO.1 ISSUE 46

JULY/AUGUST 1980

FANTASY
GAMES ISSUE!



AIR TRAFFIC



In **Air Traffic Controller** you assume responsibility for the safe flow of

air traffic within a 15x25 mile area up to 5,000 feet in altitude. During your shift as a controller in charge of this airspace, 26 aircraft become active and under your control. Jets and prop planes have to be guided to and from the two airports, navigational beacons and ten entry/exit fixes. The aircraft enter the controller's airspace at various altitudes and headings whether or not you are ready.

Air Traffic Controller retains the basic realism of air traffic control. This program requires the same steady nerves under pressure and the same instant, almost instinctive, analyses of complex emergencies which are demanded of a professional air traffic controller. But "**ATC**" adds the excitement and well-defined goals of a game. This is just a simulation, and all passengers left in air traffic limbo by a panicked player will live to fly another day.

Your goal is to get all of the aircraft to their assigned destination before the shift is completed. At your disposal are a radar display of the aircraft positions in the control area; coded information concerning aircraft heading, destination and fuel supply; nav aids enabling you to hold aircraft or assign them automatic approaches; and commands to alter the altitude and heading of the aircraft. Working against you are altitude and heading requirements, fuel restrictions and, of course, the inimitable clock.

CONTROLLER

The most obvious measure of difficulty of a game is the clock setting at the beginning. In a 99 minute game you will have time to go fix a sandwich between the appearance of two successive aircraft, while in the 16 minute game you may not have time to swallow before all of the aircraft have appeared.

No two games, even at the same clock setting, are ever alike. As controller, you must cope with the unique requirements of each aircraft. The game will end if you commit a "boundary error," that is, if an aircraft fails to leave your area at the proper altitude and exit fix...causing an unpleasant surprise for the controller next door. The game also ends if you fail to leave a comfortable margin of "safety be-

tween the aircraft as they whiz past each other. In cases of excessive delay, fuel supply considerations will become invested with a particular sense of urgency.

Successful guidance of all aircraft to their destination is a heady accomplishment. This never fails to thrill ATC enthusiasts at each successive level of play.

Your local retail store should carry **Creative Computing Software**. If your favorite retailer does not carry the software you need, have him call in your order to (800) 631-8112. Or you can order directly from **Creative Computing**. Write to **Creative Computing Software**, P.O. Box 789-M, Morristown, NJ 07960. **Air Traffic Controller** is now available for the 16K TRS-80 (3006), for the 16K Apple II and Apple II Plus (4008), the 8K Sorcerer (5008) and for the 4K Sol-20 (8001). All are on cassette for \$9.95. Include \$1.00 for postage and handling. For faster service, call in your bank card order toll free on our order hotline, (800) 631-8112.

Prices are subject to change without notice.

**creative
computing
software**

sensational software



Our focus this issue is Computer Fantasy and Simulation Games. We have so much material that we've had to skimp on space for some of our regular features. CFSG's have been a recurrent theme in *Recreational Computing* beginning with an article Lee Hoevel and I wrote for the March-April 1978 issue titled "Epic Computer Games." Since that time we've seen these games migrate from large machines to small and become widely available from commercial suppliers. John Morrison provides an overview of the CFSG's today and provides some history, background, and pointers to the literature. "Wizard's Castle" by Joseph E. Power and "The Wonderful World of Eamon" by Donald R. Brown are complete CFSG's. Both are worthy of study and play. Furman Smith has provided us with a tutorial on making your own CFSG. His program is really designed to be modified and extended into another game of your own construction. Serg Korem tackles the problem of making a dialogue based Star Trek game in a tutorial. Lastly, I have a short note speculating on the impact future technology might have on CFSG's.

NCC—the computer industry held its giant gala trade show in Anaheim towards the end of May. It was a major success if one counts heads; there were over 80,000 people there. On occasion one could not negotiate the aisles forward, backwards, sideways, or whatever.

It used to be that the personal computer people had a separate identity from the "real" computer people. This year that distinction seems to have disappeared; one found the personal computer people and the old guard mini and maxi people side by side in the exhibit hall.

We're still looking for new authors, new ideas. Your letters and comments are always welcome.

We made a couple of printing gaffs in the last issue. Some of the material printed with a color graphic over was difficult if not impossible to read. Many readers were kind enough to point it out; one even mentioned that he thought that *Imaging Age* was the only magazine to do that sort of thing and was sad to see us trying to compete with them. We will be more careful in the future. Jim Day's article, "A Proposed Graphics Language" is reprinted in this issue in a readable form. Our apologies to those of you who were outraged!

ADVERTISING

ADVERTISING space is available in this publication. Please direct inquiries to the Advertising Director, People's Computer Company, Box E, Menlo Park, CA 94025. (415) 323-3111.

SUBSCRIPTION INFORMATION

U.S. RATES

- ☐ \$12/yr. (6 issues)
- ☐ Resubscribing \$25
- ☐ \$13 tax deduction
- ☐ Resubscribing \$6100+
- ☐ \$86 + tax deduction
- ☐ Lifetime subscription \$9000+
- ☐ \$700 tax deduction

Please allow 6-8 weeks for your first issue to arrive.

BACK ISSUES

\$2.00 each:
Vol. 7, Nos. 1, 2, 3, 4, 5, 6
Outside the U.S. add \$5.00 per issue

- ☐ Payments must be in U.S. dollars on a U.S. bank
- ☐ \$20 Canada First Class
- ☐ \$26 Rest of World Airmail
- ☐ \$16 World Surface Mail

\$3.00 each:
Vol. 7, Nos. 1, 2, 3, 4, 5, 6
Vol. 8, Nos. 1, 2, 3, 4, 5, 6

Staff

Editor
Dennis Allison
Contributing Editors
Bob Albrecht
Tracy Deliman
Don Inman
Ramon Zamora
Copy Editor
Sara Le Forge
Publisher
Ann Merchberger
Art Director/
Production Manager
Carole Cullenbine
Artists
Fred Carpenter
Carolyn Miller
Carol Sevilla
Typesetters
Phyllis Adams
Renny Wiggins
Proofreader
Nancy Heubach
Marketing Director
Gavin W. Cullen
Advertising Director
Michael Madaj
Circulation Manager
Peter Clark
Circulation Assistant
Leah Dansby
Accounting Supervisor
Giuliana Peter

Retaining Subscribers

Algorithmics, Inc.
DeWitt S. Brown
Gerald Bowman
Robert Connors
David R. Dick
Mark Elgin
Joi Ellis
John B. Fried
Scott B. Guthery
Alan Hamilton
Brian Herring
T. Alton Howard
William G. Hutchison, Jr.
W. A. Kelley
William M. Richman II
Phillip A. Smith
Neil Sullivan
Joseph A. Weisbecker
Brett Wilson

Sustaining Subscribers

Bill Godbout Electronics
Byte Publications
Paul, Lori and Tom Calhoun
Louis R. Patzke



Cover by Fred Carpenter

Recreational COMPUTING

July/August 1980
Volume 9, Issue 1

FANTASY GAMES ISSUE

- 4 WRITE YOUR OWN COMPUTER FANTASY SIMULATION
by Dr. Furman H. Smith
- 10 WIZARD'S CASTLE
by Joseph R. Power
- 18 YOU, TOO, CAN FIGHT DRAGONS
by John M. Morrison
- 23 STAR TREK, A DIALOGUE APPROACH
by Serg Korem
- 27 ON FUTURE FANTASY GAMES
by Dennis Allison
- 30 CRYPTARITHMS
by John Davenport Crehore
- 32 THE WONDERFUL WORLD OF EAMON
by Donald Brown
- 46 CRYPTARITHMS FROM WAY BACK
by John Davenport Crehore
- 48 TRS-80 ART?
by Ken Hahn
- 50 MACHINES OF LOVING GRACE — A CELEBRATION
by David Thornburg
- 51 IN DEFENSE OF HACKERS
by Steven Kudlak
- 52 TOUCH, PANELS AND INTERACTIVE GRAPHICS
by Jim Day
- 54 ANNOUNCEMENTS
- 56 A PROPOSED GRAPHICS LANGUAGE
by Jim Day
- 58 PROGRAMMING PROBLEMS & SOLUTIONS
by Bob Albrecht, Don Albers & Jim Conlan
- 62 FORTRAN MAN
by L. Schneider & T. Voros

Recreational Computing (ISSN #0164-5846) is published bimonthly by People's Computer Company, 1263 El Camino Real, Box E, Menlo Park, CA 94025. People's Computer Company is a tax-exempt, non-profit corporation. Donations are tax-deductible. Second class postage paid at Menlo Park, California, and additional entry points. Copyright 1980 by People's Computer Company, Menlo Park, California.

FOREIGN DISTRIBUTORS OF *RECREATIONAL COMPUTING*: UK & EUROPE: LP Enterprises, 8/11 Cambridge House, Cambridge Road, Barking, Essex, IG11 8NT, Great Britain. Hofacker Verlag, Tegernsee Strasse 18, D-8150 Holzkirchen, West Germany. Computerland/Computer Star, AB, Box 7134, Kungälvägen 19, S-10387 Stockholm, Sweden. CANADA: RS-232, 186 Queen Street W., Toronto, Ontario M5V 1Z1, Canada. ASIA & AUSTRALIA: Electronic Concepts Pty Ltd, 55 Clarence Street, Sydney, NSW 2000, Australia. Computer Store, POB 31-261, 22 B Milford Road, Milford, Auckland 9, New Zealand. ASCII Publishing, 305 H1 Torio, 5-6-7 Minami Aoyama, Minato-ku, Tokyo 107, Japan. Systems Formulate Corporation, Shin-makicho Building, 1-8-17 Yaezu, Chuo-ku, Tokyo 103, Japan. AFRICA: Computers, Etc., 170 Hendrik Verwoerd Drive, POB 51978, Randburg 2125, Republic of South Africa.

BY DR. FURMAN H. SMITH

Invitation

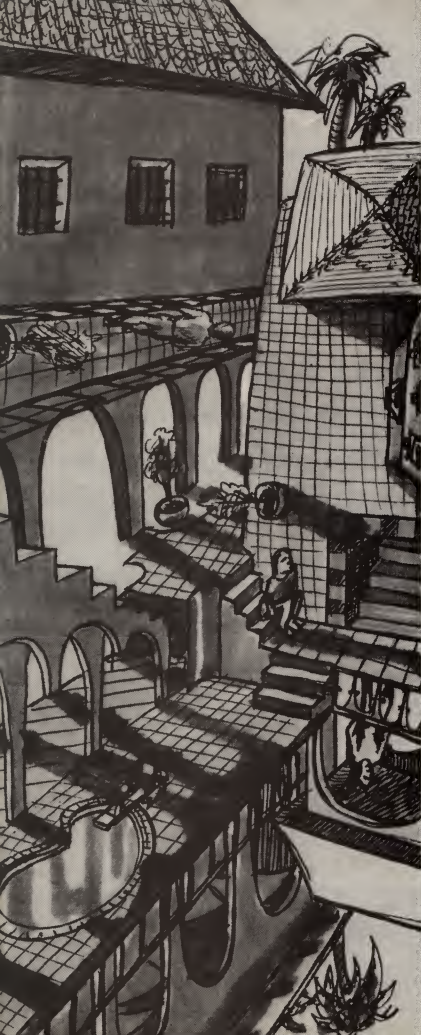
The *Nellan is Thirsty* version of *The Enchanted House* computerized fantasy simulation (CFS) is intended for children to play. The program *Enchanted House* is general and was designed for easy modification. The present article describes the program and gives ways of making simple modifications; a later article will go into major changes.

Flavor of this CFS

As with any CFS, obstacles and unfavorable situations must be overcome to win. The *Nellan is Thirsty* version is nonviolent and the creatures are all friendly; thus, planning, problem solving, and inventiveness are encouraged in a nonthreatening atmosphere.

The player indicates choice of actions by messages such as GO NORTH, UNLOCK THE DOOR, TAKE THE EGG. Only the first and last words of the messages are considered; short forms such as SOUTH and S are accepted. There are HELP, RELOOK, and LIST WORD commands. Possible commands are often suggested by all-cap words in the descriptions; for example, "There is a magic MAP here. If you GET the MAP, you may later CONSULT the MAP." In this article, I will follow the convention of capitalizing every letter of words that the program understands and messages from humans.

The magic MAP draws itself: explored rooms are labeled, their walls and doorways appear on the MAP (untried doorways as asterisks, previously tried doorways as open or blocked, according to their state on the last attempted passage); unexplored areas are shown by question marks. (While drawing a map associated with playing a CFS is a good learning experience, it is beyond the capabilities of many children who would otherwise be able to enjoy and benefit from the game; a finished map would rob them of some of the joy of exploration.)



your own computer fantasy simulation

The pathways are rectangular and usually intuitive; that is, movement is N, S, E, W, U, and D and if going NORTH from the Bank Room takes us to the White Room, then going SOUTH from the White Room will take us to the Bank (provided the passage is open). Some passages may be one-way; there may be poles that we can slide DOWN but not climb UP. Of course, magic also provides a means of transportation in the Enchanted House.

The adventurer's name is used often. If Everett is in the Bank, a giant carpet on the floor says "WELCOME EVERETT." CHULA, the white RABBIT, will greet Armenta with, "Well, my friend ARMENTA, I hope that you are enjoying your adventure."

The team of adventurers is to earn points by DEPOSITING (or DROPPing) treasures in the Bank Room. If there is more than one adventurer, each gets a turn of, for example, four scoots (scoot = one passage from one room to another). A dock in the Bank announces the time (in scoots) and a player may CONSULT her or his WATCH at any time. The game ends when all the treasures have been DEPOSITed or a certain number of scoots have elapsed.

Major Program Ideas and Notation

Things like the MAP, CAT, and COIN would usually be termed objects, but I chose to call them manifestations since some objects appear in several forms; for example, manifestation number 13 is a CAT giving the adventurer a VASE and manifestation 14 is the same CAT napping. The list of manifestations is given on lines 6200-6299 of the program.

The location of manifestation M is denoted by L(M). The information that the COUPON (M = 2) is in room 5 (the Hot Room) is summarized by

L(2) = 5. The value minus one is used when the adventurer is carrying the object; if the adventurer is carrying the COUPON, then L(2) = -1. The value zero is used when a manifestation doesn't currently exist. For example, if the BOWL with cold MILK (M = 4) is taken in the Hot Room, the location of manifestation 4 is changed from -1 to 0 since the MILK would no longer be cold, and the location of the BOWL of warm MILK (M = 5) is changed from 0 (not existing) to -1 (being carried); this transformation is handled by line 2063 of the program. The 0 and -1 conventions for L(M) are from Scott Adams' 1979 article in *Creative Computing*.

When the adventurer gives his message (example: GRAB THE COIN), the message is split into a verb part (V\$ = "GRAB") and a noun part (N\$ = "COIN"). For reasons of space and efficiency, numbers V and N, rather than character strings V\$ and N\$, are used to process the message. Reference to the list of verbs, nouns, and manifestations (lines 6200-6299) shows that "GRAB THE COIN" yields V = 3 and N = 18.

Now let's look at the listing of variables (lines 6000-6199). The number of verbs that the program understands is denoted by V9. Leaving V9 as a parameter (as opposed to a constant such as 20) makes the program easier to modify. Similarly, N9 is the number of nouns the program understands and M9 is the number of manifestations found in the game.

The "9" following a variable means (in the system of notation I've no doubt pirated from some source I've forgotten) the upper limit on that variable; thus if C denoted the number of objects that the adventurer is carrying, C9 denoted the maximum number of objects that can be carried. The current time is denoted by T and the game is over if T = T9. Don't look at line 6078 yet—what's the notation for the number of rooms in the house?

I will leave the W-vectors for the next article, but if anyone wants to build his/her own Enchanted House right away, he/she should study the block 1600-1799 in conjunction with lines 6034-6056.

Some nouns describe more than one manifestation; thus, there cannot be a one-to-one correspondence between nouns and manifestations. Line 6122 tells us that $N1(M) = N$ means that the noun number N describes the manifestation number M, for examples, $N1(2) = 13$, $N1(3) = 12$, $N1(4) = 12$.

Sometimes we have more than one noun for a single manifestation; BANKER (N = 23) and KLINKOYN (N = 130) both describe the ninth manifestation. This difficulty is overcome in general by transforming to the lowest appropriate value and in particular by changing an N of 30 to an N of 23. The same trick is used for verbs; for example, a V of 5, 6, 7, or 8 will be transformed into a V of 5.

Minor Remodeling of the Enchanted House

The program *Enchanted House* was designed so that it would be easy to build other Enchanted Houses. However, there is the worry expressed in the superb Lebling, Blank, and Anderson article, "Zork: A Computerized Fantasy Simulation Game" (July-August 1979 issue of *Recreational Computing*), although in a different context: "There is no such thing as a small change." I hope that the *Enchanted House* program allows minor changes, and I invite you to make several changes before we study major changes in a forthcoming issue.

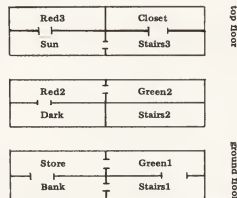
Perhaps you'd like to rename a character. It only takes a change in line 1230; for example, to name the CAT "KAIFA," replace "NELLAN" in line 1230 by "KAIFA."

Are you a stamp collector? Change the word "VASE" to "STAMP" in line 1225 and change the description in the 2649-2658 block. If the STAMP can be said to be small, that's all the modification you need make; otherwise, the only other thing that you need to do is to delete the word "small" in line 2536.

Homework for the Intrepid

Another CFS, *Deposit the Chair*, is described below. A forthcoming issue of *Recreational Computing* will have an article on modifying the current program to implement *Deposit the Chair*. The point of presenting the CFS an issue early is that some of you might want to take the modification (of the program in this issue) as a problem to work before seeing a solution in the next issue.

The *Deposit the Chair* version uses the following three-storey two-by-two map:



A CHAIR worth 30 points is initially in the Dark Room. The Red3 Room has the top of a red fireperson's pole and Red2 has the bottom of the pole. Similarly, the green rooms house a green pole. The adventurer can slide DOWN a pole under any conditions but can climb

UP if and only if she or he is not carrying the CHAIR. Similarly, the CHAIR doesn't fit through the small doorway between Green1 and Stairs1. The DOOR between the Store and Green1 is locked. One needs a SUPERBROOM to get past the COBWEB between Red1 and Green2.

The SUPERBROOM is in the Closet and the KEY is obtained by DROPPING the COUPON on the Store. Green2 is a nice initial location for the COUPON; put the COIN and JEWEL anyplace except the Bank.

I used the following values for V, N, and M. The first four manifestations are (1) machine with KEY, (2) machine without (4) Green2 Room side of COBWEB; the remaining manifestations (of M9 = 11) are identified by the noun list below and the fact that for M >= 5, N1(M) = M+14. The V9=17 verbs, in order, are TAKE, GET, GRAB, PICK, DROP, GIVE, OFFER, DEPOSIT, OPEN, UNLOCK, LIST, WORTH, HELP, GO, RELOOK, CONSULT, SWEEP. The N9 =

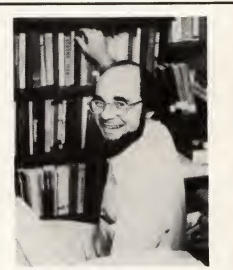
25 nouns, in order, are: N, S, E, W, NORTH, SOUTH, EAST, WEST, U, D, UP, DOWN, INVENTORY, SCORE, WORDS, WATCH, DOOR, COBWEB, COUPON, KEY, SUPERBROOM, MAP, COIN, JEWEL, CHAIR.

If you attempt this exercise, be sparing in your descriptions. Text takes a lot of memory.

Documentation and Dissemination

Documentation is very expensive in terms of storing the program. My 48k system will not hold both BASIC and the program printed in this article. Fortunately, I have a single disk system (a Heath H89 "All-in-One") and there's plenty of room on a disk. I wrote a program I call *Hack and Pack* which takes a program as input, hacks out all comments, deletes unnecessary blanks, and thus produces an output program that is difficult to read but which can be stored in much less space. An example line of output is found in line 35 of the program printed here.

Furman Smith received a Ph.D. in Probability and Statistics from the Florida State University in 1972, taught three years at the University of Kentucky, and has since been at the University of Houston Victoria Campus. He is currently an Associate Professor of Mathematical Sciences teaching four courses, Chairperson of the UHUC Faculty Council, and a member of numerous committees including an eight faculty member group that is advisory to the President of the University of Houston System. He has a marvelous wife, two marvelous kids, one good home computer, a garden, and backlog of work.



Enchanted House

Nellie is Thirsty

Furman W. Smith
University of Houston Victoria Campus
2382-C Red River
Victoria, Texas 77981

```

12  This program appears in the
13  July-August 1988 issue of RECREATIONAL COMPUTING.
14  Copies for the Radio Shack level II 16 computer-k TRS-80
15  will be available from RECREATIONAL COMPUTING.
16  Corv submitted to Heath User's Group for MOOS MHSIC
17  (c) users with 16 computer-k bytes above MHSIC.
18  Warning! The program as written here will not fit in
19  an 48 computer-k system. The program that I run has
20  all comments and unnecessary blanks deleted; for ex.,
21  this line isn't in the program and line 135 looks like
22  135 135 IFPS:THEPRINTON, come on, 15070130
23  Radio Shack users should omit the THEN in such lines.
24  8- 99 assigns credits and contents
25  100-199 initializes
26  200-299 describing locations & manifestations
27  300-399 determining player's actions
28  400-499 ordinary savings, settings, & droppings
29  500-599 otherwise responding
30  600-699 updating
31  700- 799 instructing player
32  800- 899 (blank)
33  1000-1999 details of initializations. In particular:
34  100x introducing values of parameters & DIM
35  110x introducing verbs
36  120x introducing nouns
37  130x intro. graphics for rectangular floor plan
38  140x intro. graphics for possible doorways
39  150x introducing rooms
40  160x describing N-S doorways
41  170x describing E-W doorways
42  180x certain properties of manifestations
43  190x initializations other than above
44  2000-2999 details of describing rooms
45  2100-2999 details of describing manifestations
46  3000-3999 responding to unlisted words
47  4000-4999 details of ordinary moving, settings, & droppings
48  4100-4999 drawing the map
49  5000-5999 details of otherwise responding
50  6000-6199 listing variables found in the program --- note
51  6200-6399 listing verbs, nouns, & manifestations --- note
52  ----- initializing -----
53  101  What does line 105 do? See lines 186-189.
54  105  CLEAR 250 105SUB 705 105SUB 1010 105SUB 745
55  106  CLEAR 250  allocates extra string space
56  107  GOSUB 705  puts some instructions on the screen
57  108  GOSUB 1010  rushes off & initializes
58  109  GOSUB 745  hops back & interacts with the human
59  110 INPUT "Are you the only person playing (YES or NO) " : RS
60  115 P = 1 : RS = LEFT$(RS,1) : IF RS="Y" THEN P9 = 1 : GOTO 140
  
```



```

1460 DATA "++-+++", "B-B"
1461 doorways that were blocked last time tried
1462
1470 Lines 1470-1475 are for the Health H19 & H99.
1471
1472 U1(0) = VGS(1470)++H99 & U2(0) = H99 & "++H99
1473 U1(1) = VGS(1471)++H99 & U2(1) = VGS(1)++H99
1474 U1(2) = VGS(1472)++H99 & U2(2) = VGS(2)++H99
1475 U1(3) = VGS(1473)++H99 & U2(3) = VGS(3)++H99
1476
1477 Lines 1482-1487 are for Radio Shack level II TRS-80
1478 considers with over 16 channels of RAM.
1479
1480 R# = CHR(140) + CHR(140) + CHR(140)
1481
1482 U1(0) = R# + CHR(140) + U2(0) = "
1483 U1(1) = R# + LEFT$(R#,2) + U2(1) = CHR(170) + CHR(149)
1484 U1(2) = R# + CHR(140) + U2(2) = "
1485 U1(3) = R# + CHR(152) + CHR(164) + U2(3) = CHR(162) + CHR(132)
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```


USEFUL TALK FROM THE DOCTOR.

New! DR. DOBB'S JOURNAL OF COMPUTER CALISTHENICS & ORTHODONTIA: Running Light Without Overbyte Vols. 1, 2, & 3

(The People's Computer Company Series). Each volume includes almost everything from all of the issues of Dr. Dobb's Journal published in the years indicated.

Vol. 1 contains the first ten issues and reflects the changes that took place in personal computing in 1976. Chapters include: A Realized Fantasy; Unlimited Speech Synthesis for Home Computers; A 6502 Disassembler from Apple; Timer Routines for 6800's; A BASIC Terminal Exchange Program; CHASE: A One or Two Player Video Game, and more.

Vol. 2 (1977) chronicles the emergence of the small computer as a useful tool. It includes the source listing for Lawrence Livermore BASIC, the source code for Dr. Starkweather's 8080 Pilot, and much more. Chapters include: A KIM-1 TV Sketch Program; What to Get Your Computer for its First Birthday; More Support for Software Vendors; Some Dire Warnings of a Modern, and more.



Vol. 3 (1978) looks at the growing interest in programming languages, along with articles on specialized applications and utilities. Chapters include: Memory Test for 6502; Relocation is Not Just Moving Programs; Z-80 RAM Tester; Absolutely Dazzling Dazzleware; Renumbering and Appending BASIC Programs on the Apple-II Computer; STRUBAL: A Structured BASIC Compiler; A KIM Binary Calculator; SAM 76 — Additional Comments, and more. **Vol. 1, #5475-0; Vol. 2, #5484-X; Vol. 3, #5490-4; \$18.95 each.**

Available at your local computer store!



HAYDEN BOOK COMPANY, INC.
50 Essex Street, Rochelle Park, N.J. 07662

**Call (201) 843-0550, ext. 307
TO CHARGE YOUR ORDER TO
Master Charge or BankAmericard!
Minimum order is \$10.00;
customer pays postage and handling.**



WIZARD'S CASTLE

BY JOSEPH R. POWER

Wizard's Castle is written for a 16K or larger Exidy Sorcerer. It is a fantasy game about equal in complexity to the many ADVENTURES now available and is the largest program I have ever written in BASIC. Unlike ADVENTURE, the castle is randomly stocked for each game, which means that winning one does not detract from future games. At present, there are versions of Wizard's Castle for the Sorcerer, the TRS-80, and HP 2000 BASIC.

BACKGROUND HISTORY

Many cycles ago, in the kingdom of N'dic, the gnomish wizard Zot forged his great orb of power. Soon after this he vanished, leaving behind his vast subterranean castle filled with esurient monsters, fabulous treasures, and the incredible Orb of Zot. From that time hence many a bold youth has ventured into the wizard's castle. As yet, none has ever emerged victorious.

INTRODUCTION

Wizard's Castle is a computerized simulation of one of the most common and popular fantasy motifs—the lone adventurer's quest within an immense underground labyrinth. Each game is separate from all others, so the game is a challenge even after you have won several times. Each game will result in a win or loss, depending on a player's skill and luck. The following instructions explain the rules and options of the game. If at any time, however, you are not sure of what to do—experiment. The program is designed to prevent invalid inputs.

CHARACTER CREATION

At the start of each game you will be asked a number of questions about what type of character you will have. You must make choices about the following things:

Race—You may be an Elf, Dwarf, Man, or Hobbit. Each race starts with a total of 32 points (except Hobbits, who get 28) but they are distributed differently for each race.

Sex—You may be male or female. Both are equal in ability and number of points. Be creative with your response.

Points—Each character starts with a number of points for the attributes of Strength (ST), Intelligence (IQ), and Dexterity (DX). In addition, there are some other points which you may distribute to the three attributes as you wish.

Your ST, IQ, and DX may be any number from 1 to 18. If any of the three goes below 1, you have died. For all three attributes the bigger numbers are better. Each character also starts the game with 60 gold pieces (GPs) with which to purchase some, none, or all of the following:

Armor—You may buy Plate armor for 30 GPs, Chainmail for 20 GPs, or Leather for 10 GPs. You can wear only one suit of armor at a time. The more expensive the armor, the more damage it will absorb.

Weapons—You may buy a Sword for 30 GPs, a Mace for 20 GPs, or a Dagger for 10 GPs. You can carry only one weapon at a time. The more expensive the weapon, the more damage it will do to the various monsters.

Lamp—If, after selecting armor and a weapon, you have 20 GPs or more left, you may buy a lamp for 20 GPs. Having a lamp will allow you to look into adjacent rooms without

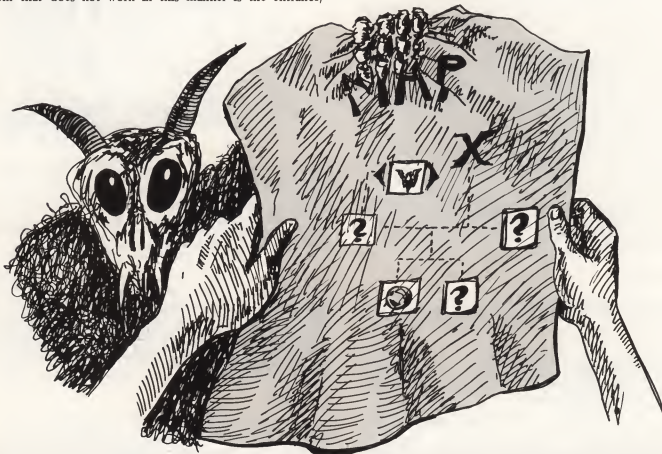
entering them.

Flares — If, after all other purchases, you have any money left, you may buy flares at 1 GP apiece. Lighting a flare reveals the contents of all the rooms around your current position.

Once you have created and equipped a character, you are ready to enter the castle and begin the game.

THE CASTLE

The castle is arranged as an 8 x 8 matrix. This means that there are 8 levels with 64 rooms in each level. Each level of the castle is constructed like a donut in that the north edge is connected to the south edge and the east edge is connected to the west edge. In a similar fashion, the sinkholes (explained later) on level 8 will 'drop' you down to level 1. The ONLY room that does not work in this manner is the entrance/



exit of the castle (always at (1, 4) Level 1). Going north from this room takes you out of the castle and ends the game.

Each room in the castle will have as contents one of the following:

- E — the entrance/exit of the castle, always located at (1,4) Level 1
- ♦ — an empty room
- U — stairs going up to the level above
- D — stairs going down to the level below
- P — a magic pool you can drink from
- C — a chest which may be opened
- B — a book which may be opened
- G — from 1 to 10 gold pieces
- F — from 1 to 3 flares
- W — a warp
- M — a monster (1 of 12 types)
- V — a vendor

- O — a crystal orb (this is NOT the Orb of Zot)
- S — a sinkhole (a room with no floor)
- T — one of the eight treasures.

The single letters are the abbreviations for the room contents which are displayed whenever you look at your map or light a flare. When you look at your maps, the room you are currently in is bracketed by <>.

PLAYER COMMANDS

Whenever the program asks YOUR MOVE?, you must decide what action you wish to perform. If your choice is not valid in your current situation, the program will inform you and allow you to try again. The following is a list of the commands which the program understands, with a description of their effects and restrictions:

NORTH moves you to the room north of your present position. WHEN YOU GO NORTH FROM THE ENTRANCE THE GAME ENDS. In all other cases the north edge wraps to the south.

SOUTH moves you to the room south of your present position. In all cases the south edge wraps to the north edge.

EAST moves you to the room east of your present position. In all cases the east edge wraps to the west edge.

WEST moves you to the room west of your present position. In all cases the west edge wraps to the east edge.

UP causes you to ascend stairs going up. You must be in a room with stairs going up.

DOWN causes you to descend stairs going down. You must be in a room with stairs going down.

DRINK causes you to take a drink from a magic pool. You may repeat this command as often as you wish, but you must be in a room with a magic pool.

MAP causes a map of the level you are currently on to be printed. All unexplored rooms are displayed as '?'; all other

rooms are displayed by their one-character abbreviations. You may look at your map at any time unless you are blind or retreating from a monster.

FLARE causes one of your flares to be lit, revealing the contents of all the rooms around your current position. Because each edge is joined to the opposite edge, you will always see nine rooms with your location in the center. Each flare you light burns away to nothing. You may light flares as long as you have some and if you are not blind or retreating from a monster. For people familiar with any of the common Star Trek games, a flare works much like a Long Range Scan with the addition of wraparound.

LAMP allows you to shine your lamp into any one of the rooms north, south, east, or west of your current position, revealing that room's contents. Unlike flares, a lamp may be used over and over again. You may use your lamp at any time unless you don't have a lamp, are blind, or are retreating from a monster.

OPEN causes you to open the book or chest in the room you are in. This command will only work if you are in a room with a chest or book.

GAZE causes you to gaze into a crystal orb. When you see yourself in a bloody heap, you lose 1 or 2 points of Strength. When you see the location of the Orb of Zot, there is only a 50% chance that it is correct. You cannot gaze when you are blind or when you are not in a room with crystal orb.

TELEPORT allows you to teleport directly to a room. This is the only way to enter the room containing the Orb of Zot. You must have the Runestaff to teleport.

QUIT allows you to end the game while still in the castle. You will be asked if you really want to quit, in case you made a mistake or want to change your mind. If you quit, you will lose the game.

All of the commands, except **DRINK**, can be abbreviated by their first letter (M for MAP, Q for QUIT, etc.). The abbreviation for **DRINK** is DR (to avoid confusion with the **DOWN** command).

MAGIC SPELLS

Whenever your Intelligence (IQ) becomes 15 or higher, you gain the option to cast a magic spell on a monster if you have the very first combat option. The three spells and their effects are:

Web traps the monster in a sticky web so it can't fight back as you swing at it. This spell lasts between 2 and 9 turns and costs you 1 Strength (ST) point.

Fireball hits the monster with a ball of flame that causes between 2 and 14 points of damage instantly. It costs 1 ST point and 1 IQ point.

Deathspell is a contest of wills between the monster and yourself. Whoever has the lower IQ dies at once. It costs nothing to use but it is very risky. Even with an IQ of 18 (the highest possible), you have a 25% chance of losing.

TREASURES, CURSES, BLINDNESS, AND SUCH

In the castle are eight randomly placed treasures:

The Ruby Red — wards off the curse of Lethargy.

The Norn Stone — has no special power.

The Pale Pearl — wards off the curse of the Leech.

The Opal Eye — cures blindness.

The Green Gem — wards off the curse of Forgetfulness.

The Blue Flame — dissolves books stuck to your hands.

The Palantir — has no special power.

The Silmaril — has no special power.

There are three curses:



Lethargy — this gives the monsters the first attack which prevents you from bribing them or casting spells on them.

Leech — this takes from 1 to 5 GPs from you each turn until you have no more.

Forgetfulness — this causes you to forget what you know about the castle. Your map slowly returns to all question marks; however, the room contents remain the same.

In addition to nullifying the effects of the curses, the treasures also provide protection from the two bad things that can happen when you open a book. These are going blind, which prevents you from seeing your maps, lighting flares, using your lamp, gazing into orbs, and being told your present location, and having the book stick to your hands, which prevents you from using your weapon to fight monsters (you can still cast spells at them though).

VENDORS

On every level in the castle there are vendors who are more than willing to sell you various items at grossly inflated prices. Normally, the vendors will make you an offer for every treasure you have and then, depending on the amount of gold you have, will sell you new armor, a new weapon, ST, IQ, and DX potions (no matter how many of these you buy, 18 is the maximum value for all three), and a lamp if you don't already have one.

If you choose to attack a vendor, you will antagonize every vendor in the castle and they will react like a monster. You will also lose the ability to trade with them. Killing a vendor, however, will give you new plate armor, a sword, one ST, IQ, and DX potion, and a lamp (if you don't have one), in addition to his hoard of between 1 and 1000 GPs.

To end hostilities and reestablish trade, you must bribe any vendor in the castle with the treasure of his choice.

MONSTERS AND THE RUNESTAFF

There are twelve types of monsters in the castle:

- | | |
|-----------|-------------|
| 1. Kobold | 7. Bear |
| 2. Orc | 8. Minotaur |
| 3. Wolf | 9. Gargoyle |
| 4. Goblin | 10. Chimera |
| 5. Ogre | 11. Balrog |
| 6. Troll | 12. Dragon |

Please note that each time you hit a gargoyle or dragon there is a chance your weapon will break.

Each monster possesses a hoard of from 1 to 1000 GPs which you get when you kill it. In addition, one of the monsters is carrying the Runestaff (you won't know which one until you kill it). You must have the Runestaff to teleport, and when you teleport into the room containing the Orb of Zot, the Runestaff disappears.

WARPS AND THE ORB OF ZOT

All but one of the rooms labeled W really are warps, and walking, falling, or teleporting into them will cause you to move to any room in the castle at random. The one exception is the Orb of Zot which is disguised as a warp. Walking into this room always causes you to move one room further in the same direction. To enter the room, you must teleport in. At this point you acquire the Orb of Zot and the Runestaff vanishes. To win the game, you must then leave the castle with the Orb of Zot. Remember that crystal orbs are not the Orb of Zot and cannot be picked up.

ERROR MESSAGES

Any time you receive a message starting with ** it means that the last thing you typed was unacceptable to the program at that time. For instance, if you get the message ** NO ORB - NO GAZE, this means that you tried to gaze in a room that did not have an orb in it. You are always required to redo your last response when you receive a ** message.

CREDITS

No program is created or exists in a vacuum, and Wizard's Castle is no exception. Many thanks go to Chip Bestler for creating the first Hobbit program which served as the great-grandfather to this one, to Kevin Williams and Dana Kaempfen for thinking the game was good enough to beg me for it, and to all the people who asked me for one more 'little' addition to it.



FINAL INSTRUCTIONS

These instructions are meant to be a guide only. Feel free to experiment with various responses when running the game (this is half the fun of the game). For the best results, add equal parts of imagination and common sense. If you have any questions that you can't figure out, write to me and I will do my best to dispell confusion. Send all letters to Joseph R. Power, 124 Cedar St., Apt. 5, E. Lansing, MI 48823. Please include a self-addressed stamped envelope.

Variables, Functions, and Machine Specifics for Wizard's Castle

CS(34)	names of all the possible castle contents
IS(34)	abbreviations for all the possible castle contents
RS(4)	names of the four races
WS(8)	names of the four weapons and four armor types
ES(8)	names of the eight recipes (for Orc Tacos, etc.)
C(3,4)	locations and status of the curses
T(8)	status flags for the treasures (1 = player owns)
O(3)	location of the Orb of Zot
R(3)	location of the Runestaff
X, Y, Z	usually the location coordinates of the player
book-stuck-to-hands flag (1 = book stuck)	
Vendor-anger-flag (1 = Vendors angry)	
LF	lamp-owned flag (1 = player owns it)
RF	Runestaff possession flag (1 = player owns it)
OF	Orb of Zot possession flag (1 = player owns it)
BL	blindness flag (1 = player is blind)
BL	blindness flag (1 = player is blind)
ST	current number of strength points
IQ	current number of intelligence points
DX	current number of dexterity points
OT	amount of other points the player gets
AV	number of points your armor absorbs per hit
AH	total number of hit points your armor has left
WV	number of points of damage your weapon does
GP	total number of treasures you possess
FL	total number of gold pieces you possess
HT	total number of flares you possess
HT	last turn you ate a monster on
T	the turn counter

all other variables are temporary and reusable or else string constants used in messages.

$FNA(Q) = 1 + INT(RND(8) * Q) < -$ produces a random number from 1 to Q
 $FNB(Q) = Q + 8 * ((Q - 9) - (Q - 0)) < -$ causes wraparound at borders
 $FNC(Q) = -Q * (Q < 19) - 18 * (Q > 18) < -$ limits Q to a max. of 18
 $FND(Q) = Q * 64 + X * 8 + Y - 585 < -$ computes room location in memory
 $FNE(Q) = Q + 100 * (Q > 99) < -$ tags a room as explored.

In Sorcerer BASIC TRUE = -1 and FALSE = 0.

All variables are global in a user defined function except the dummy parameter.

CHRS(12) clears the screen like a TRS-80 CLS command.

The first remark is a machine language routine to simulate the RANDOM function.

After 32767, memory locations (for POKE and PEEK commands) are numbered -32768 (8000 hex) to -1 (FFFF hex).

LISTING



"WIZARD'S CASTLE"

[illegible][illegible]

```

980 Z$="P":GOT01000
990 Z$="DOWN":PEEK(FND Z)=4:THENZ=Z+1:GOT01670
1000 PRINT:PRINT**"OH *FRKRC)"** NO STAIRS GOING *Z$*" IN HER
E:GOT0620
1010 IFBL=1:THENPRINT:PRINT**"YOU CAN'T SEE ANYTHING, DUMB *FRK
RC):GOT0620
1020 PRINT:PRINTA=X:B=Y:FORX=1:TOB:FORY=1:TOB:PEEK(FND Z)):IFD
978:THENZ=Z+4
1030 IFX=ANDY=BTHEPRINT"< *FRK 0)"**":GOT01050
1040 PRINT:PRINTA=X:B=Y:NT:KEYY=A+Y:B:GOT01100
1050 PRINT:PRINTA=X:B=Y:NT:KEYY=A+Y:B:GOT01100
1060 PRINT:LEVEL="Z":GOT0620
1070 IFEL=0:THENPRINT:PRINT**"HEY BRIGHT ONE, YOU'RE OUT OF FLAR
ES":GOT0620
1080 PRINT:PRINTFL=FL-1:IFX=B:FORI=A-1:TO11:IF(FNB 01):OR0=2=
B-1:TOB:11=PRINT:GOT2
1090 G=PEEK(FND Z)):PEEK(FND Z)):PRINT$(0):**"NEXT02:FR
INT:PRINT:TEXT01:X=Y:B
1100 GOSUB400:GOT0620
1110 IFL=0:THENPRINT:PRINT**"YOU DON'T HAVE A LAMP, *FRKRC):GOT
0620
1120 PRINT:PRINT"WHERE DO YOU SHINE THE LAMP (N,S,E, OR W) *":GOT
SUB3290
1130 Z$=X+B-Y:X=FNB(X+0$="N")-(0$="S")Y=FNB(Y+0$="W")-(0$="E")
1140 IFX=X-B-Y:0:THENPRINT:PRINT**"TURNKEY! THAT'S NOT A DIRECTIO
N":GOT0620
1150 PRINT:PRINT"THE LAMP SHINES INTO (*YX)* *Y*Y*") LEVEL *Z:FR
INT
1160 PEEK(FND Z):FNB(PEEK(FND Z)):PRINT"HERE YOU WILL FIND *IC$
(PEEK(FND Z))"
1170 X=A+Y:B=GOT0620
1180 IFPEEK(FND Z)>0:5:THENPRINT:PRINT**"IF YOU WANT A DRINK, FI
ND A POOL."GOT0620
1190 IFPEEK(FND Z)>0:5:THENPRINT:PRINT**"YOU TAKE A DRINK AND *":IFD:0:THENPRINT
"FEEL *"
1200 ONGOT01210,1220,1230,1240,1250,1260,1270,1290
1210 ST=NC(ST:FNB(3)):PRINT:ST=NC(1:GOT0620,2840
1220 ST=ST-FNB(3)):PRINT:WE=NER-1:ONL-1:GOT0620,2840
1230 IO=NC(10:FNB(3)):PRINT:SMARTER-1:GOT0620
1240 IO=NC(10:FNB(3)):PRINT:DUMBER-1:ONL-1:GOT0620,2840
1250 IO=NC(10:FNB(3)):PRINT:SMARTER-1:GOT0620,2840
1260 IO=NC(10:FNB(3)):PRINT:DUMBER-1:ONL-1:GOT0620,2840
1270 IO=NC(10:FNB(3)):PRINT:SMARTER-1:GOT0620,2840
1280 RC=0:PRINT:DECODE A *FRKRC):GOT0620
1290 SX=1-SX:PRINT"TURN INTO A *":IFSX:0:THENPRINT"FE"
1300 PRINTNAME *FRKRC):GOT0620
1310 IFPEEK(FND Z)=6:THENPRINT:PRINT"YOU OPEN THE CHEST AND *PR
INT:GOT01430
1320 IO=PEEK(FND Z)=12:THENPRINT:PRINT"YOU OPEN THE BOOK AND *PR
INT:GOT01340
1330 PRINT:PRINT**"THE ONLY THING YOU OPENED WAS YOUR BIG MOUTH
":GOT0620
1340 ONFNA(6:GOT01350,1360,1370,1380,1390,1400
1350 PRINT"FLASH! OH NO! YOU ARE NOW A BLIND *FRKRC):BL=1:GOT01
420
1360 PRINT"ITS ANOTHER VOLUME OF ZOT'S POETRY! - YEECH!":GOT0142
0
1370 PRINT"ITS AN OLD COPY OF PLAY *FRKRC):BL=1:GOT01420
1380 PRINT"ITS A MANUAL OF DEXTERITY *ID=18:GOT01420
1390 PRINT"ITS A MANUAL OF STRENGTH *I=18:GOT01420

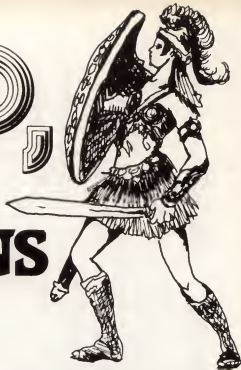
```

```

1400 PRINT"THE BOOK STICKS TO YOUR HANDS *":PRINT
1410 PRINT"NOW YOU CAN'T DRAW YOUR WEAPON *":BF=1
1420 PEEK(FND Z),1:GOT0620
1430 ONFNA(4:GOT01440,1450,1460,1470
1440 PRINT"BARBOSH IT EXPLODES *":GOSUB300:ONL-1:ST<1:GOT
01420,2840
1450 G=FNA(100):PRINT"FIN! *G" GOLD PIECES *GP=G+40:GOT01420
1460 PRINT"Gasi YOU STRAGGLER FROM THE ROOM"
1470 PEEK(FND Z),11:IF=12:0:5=NT:W"NSEW" *FNA(4),11:GOT01950
1480 IFPEEK(FND Z)>0:11:THENPRINT:PRINT**"NO ONE - NO GAME":GOT01
620
1490 PRINT:PRINT"YOU SEE *":ONFNA(6:GOT01500,1510,1530,1540,1560
1500 PRINT"YOURSELF IN A BLOODY HEAP *ST=ST-FNB(3)):ONL-1:GOT
0620,2840
1510 PRINT"YOURSELF DRINKING FROM A POOL AND BECOMING *ICK *IFN
AT(13)
1520 GOT0620
1530 PRINTICK(12:FNA(13))" GAZING BACK AT *":GOT0620
1540 A=X:1=B:Y=C:Z=FNB(8)):FNB(8):Z=FNB(8):10=FNB(PEEK(FND Z)):P
EAK(FND Z)):0
1550 PRINTICK(0)**" AT (*YX)* *Y*Y*") LEVEL *Z:Z=X+A-Y:B=Z+C:GOT0620
1560 A=FNB(8):B=FNB(8):C=FNB(8):IF(FNB(8):X)=FNB(8):1:1:0:2:1:C=0(
3)
1570 PRINT"THE ORB OF ZOT AT (*YX)* *Y*Y*") LEVEL *Z:GOT0620
1580 PRINT"THE ORB OF ZOT AT (*YX)* *Y*Y*") LEVEL *Z:GOT0620
1590 PRINT"THE ORB OF ZOT AT (*YX)* *Y*Y*") LEVEL *Z:GOT0620
1600 Z$=X-Y-C:OOR(1:1:FOR NORTH B=FOR SOUTH) *":GOSUB3350:X=0
1610 Z$=X-Y-C:OOR(1:1:FOR WEST B=FOR EAST) *":GOSUB3350:Y=0
1620 Z$=Z-LEVEL (1:1:FOR BOTTOM) *":GOSUB3350:Z=0
1630 G$=T:GOT01670
1640 PRINT:PRINT"DO YOU REALLY WANT TO QUIT *":GOSUB3290:PRINT
1650 IF0$>Y:THENPRINT"NO THEN DON'T SAY YOU DO":GOT0620
1660 PRINT:PRINT:GOT02940
1670 PRINT:IFBL=0:THENGOSUB400:PRINT
1680 PRINT"ST= *ST*:ID= *ID*:DX= *DX*:FLARES= *FL*:GP *S=
*GP"
1690 PRINT:PRINT"MW1)" *":IF(FNB(AV+5)):IFL=1:THENPRINT" / A LA
MP *"
1700 PRINT:PRINT:WC=0:0=FNB(PEEK(FND Z)):PEEK(FND Z)):01Z$="YOU H
AVE *"
1710 PRINT"HERE YOU FIND *ICK(0):IF(0:7:OR(6=11:OR(6=12:THEN620
1720 IF0:7:THENGP=GP+FNA(4):0:PRINT:PRINTZ+3:GP:GOT01420
1730 IF0:8:THENFL=FL+FNB(5):PRINT:PRINTZ+5:FL:GOT01420
1740 IF0:9:THENL70
1750 PRINT:IF(0:1=X)X=AND(0:2=Y)Y=AND(0:3=Z)Z=THENONL-1:0$="T":GOT01
950,3050
1760 X=FNB(8):Y=FNB(8):Z=FNB(8):GOT01670
1770 IF=10:THENZ=FNB(Z+1):GOT01670
1780 IF0:25=AND0<3:ATHEPRINT:PRINT"ITS NOW YOURS *IT(0-25)=1:TC=TC
+1:GOT01420
1790 A=PEEK(FND Z))-12:IF0=0:IF(A,13:OR(W=1) THENZ300
1800 PRINT:PRINT"YOU MAY TRADE WITH AN ATTACK, OR IGNORE THE VENDE
R"
1810 GOSUB3280:IF0$="T":THEN620
1820 IF0$="A":THENF=1:PRINT:PRINT"YOU'LL BE SORRY YOU DID THAT *":
GOT02300
1830 IF0$>"T":THENPRINT:PRINT**"NICE SHOT, *FRKRC):GOT01800
1840 PRINT:FORO=1:TOB:FORA=0:TO150:IF(0:1:THEN1880
1850 PRINT:PRINT"DO YOU WANT TO SELL *Y$K(0:25)* *FOR *A$*" GP *S

```


YOU, TOO, CAN FIGHT DRAGONS



BY JOHN M. MORRISON

We all know what computers were built for from their inception—to play games on.

It is an extremely unusual, and probably extremely dull as well, programmer who has not owned, run, or "modified" such programs as *Star Trek*, *Lunar Lander*, and a relative newcomer called *Adventure*. These games, particularly the last one, belong to the genre of Fantasy Role-Playing Games.

"A fantasy role-playing game is one where the players (sometimes one, often several) can assume the personas and personalities of imaginary beings—not always human—who are thus given lives and adventures of their own, directed by the game players.

In the course of the game, characters (the imaginary ones) are presented with circumstances created in outline by a Game Master. The fulfillment of that outline depends on a verbal give-and-take between the Game Master and the players as they explore the world presented, taking actions to cope with the situations in which they find themselves.

The Game Master is a real person who has taken the time to create the scenario in which his or her friends can play. The players may be using medieval knights, space-faring aliens, or a cowpuncher from the Old West."

So, a game of this sort basically consists of some fictional characters, who are controlled by players, running around in a universe devised by a person who takes the place of both God and the senses of the adventurers. This person, the Game Master, must not only create a situation but must describe what the players see, hear, etc. as if they were there themselves. They've been described in other places as "complicated forms of 'Let's Pretend'."

Is it any wonder, then, that the task of Game Master would eventually be handed to computers? After all, a computer is certainly more consistent and does not (often) forget rules. The first game of this sort, created by Don Woods and William Crowther, was *Adventure*. In this game, the player is given a description of a scene (e.g., "You are in a dry stream bed") and is then asked what he or she wishes to do. Although it might seem tame, there are homicidal dwarves that pop up to kill the unwary, and fabulous treasures that must be gathered before one can be considered an "expert."

After *Adventure* came *Zork*, which was written on the MIT DEC-10 but soon found its way onto literally hundreds of other computer systems. Essentially, *Zork* is similar to *Adventure*, but has more rooms, more complex puzzles to solve, and a better parser. (A "parser" is a routine that translates English, or other languages, to a format understandable by the machine. More on this later.) There are several others of its ilk (I have played *Haunted House* and *The Shire*, to name two), all similar in that they first describe a situation and then ask for input.

With better terminals came more games. On the PLATO terminals on many different computers are several such as *Dungeons, Monsters, and Treasure* and *Orthanc*, where a little figure is propelled through a series of corridors and rooms, and *Moria* and *Baguette* in which the player sees an actual visual representation of what the fictional character would see. In another, a joystick-controlled tour through an entire imaginary town may be taken—all, of course, in acceptable perspective.

But as any avid computer gamer in college knows, games stand very little chance of staying on-line. Administrators have an odd aversion to them. Thus... the microcomputer.

Already many of the earlier programs, notably *Adventure*, are available for many of the micros on the market. Naturally, though the advancement of the art did not stop there. Innovative folks created so many role-playing game programs that they are impossible to list here, although a good overview can be obtained by surveying advertisements in personal computer magazines.

It is the aim of many to create their own programs. Sometimes a little help is needed to get started in such projects—thus, this article.

It really doesn't matter what the adventure itself is. Although not a good tactic in writing fiction, a game set in the Old West can be changed to a one in the Far Future by changing "six-gun" to "proton blaster," etc. The manipulation of data is the most important part. A good FRP program should be able to deal with any information it is presented with. Of course, any program should have this characteristic, but role-playing games have special problems. For example, a rather common spell in the *Dungeons and Dragons* rule system is "Charm

Person," which can bring a human under the control of another. Any competent Game Master can determine whether or not the spell is successful and the extent to which the victim is controlled. But how does a computer translate these parameters into ones that it can understand?

Let's start with a simpler problem: what must be in any FRP system? Naturally, one or more *characters* are necessary; because of the way terminals are set up (one keyboard to one screen), it is easiest to implement one-character games. More players, especially with autonomy of their own, cause both the economics and the intricacy of the game to increase geometrically. There are new software packages on the market, though, that allow one to tap into larger computers, so the outlook for multiplayer games is not hopeless.

These characters need certain *abilities* to flesh them out. These may take the form of Strength, Intelligence, Dexterity, Beauty, Luck, Experience, various levels of skills, etc. Most games operate with two to eight of these characteristics, but because they can easily be put into integer variables, a larger number can be implemented on a computer. Don't, however, fall victim to the other extreme—that of having too many statistics. If REFLEX-SPEED and MANUAL-DEXTERITY are used to refer to the same thing, one can be deleted.

An *inventory* is another important data field. Characters will not be fighting their foes bare-handed all of the time (although you may wish to start them out that way); swords, guns, shields, etc., should be on sale someplace or findable within the adventure. If you allocate an array, or a number of arrays, you can easily work with any particular item. Although this limits characters to a certain number of possessions, I can really not imagine most people carrying more than twenty individual things. A more advanced method is to use pointers; in other words, the array of items would have an additional field, NEXT. The player would have a variable indicating the number of this first item, and the next item would be the number in NEXT of the previous item. If NEXT is negative (or some other key value) that item is the last in the character's list. This method also lets items own other items such as a backpack or a duffel bag. In large dungeons this can save quite a lot of space. Gold, or other monetary units, is also handy.



The program deals with manipulating characteristics and possessions in various environments. Two types of program structure are evident at this point: "hard" and "soft" (as in hardware and software).

Adventure is an example of the hard-type. The room descriptions are enclosed within the statements of the program. A more extreme example would be one where, every so often (a hundred lines or so) a new room starts and is self-contained within a block. This happens most often when one writes programs at the terminal, thinking them out as one goes along. Except for very small, very restricted varieties, it is not recommended.

Of course, there are exceptions to all rules. It seems that if you can take adventures and break them down into sub-units (for example, in one named DRAGONQUEST the first subadventure may be ROOM-1, which is further broken down into TAKE-SWORD, READ-BOOK, and EXIT-TO-EAST), you can run them using a simple recursive procedure that might go



```
ADVENTURE (name):
INPUT COMMAND
IF COMMAND = option #1 THEN ADVENTURE (sub-
  adventure #1)
IF COMMAND = option #2 THEN ADVENTURE (sub-
  adventure #2)
```

```

      *
list of other options
      *
etc.
```

The list of commands, of course, would include basic ones other than "options" (this idea of the "command loop" will be covered later). Once a command is executed, the next higher level of recursion would take over. This is a useful tool when a subadventure can take place in a number of locations, for example, a book that can be read anywhere.

The soft type has many advantages, one of which is that by changing the data a totally new adventure can be created (especially if read in from a file). In fact, you may have to write an auxiliary program to input the data. Of course, watch who you give the "editor" program to, as anyone who has it can easily cheat in the actual play of the adventure!

Commands may also be hard and soft. A 'command' is, for example, "go north," "fight the hobbit," "tear page 7 from the textbook;" in other words, telling the controlling device what you want your character to do. Some of the more common commands are the following:

Command	Abbreviation	Explanation
TAKE	T	If the specified item is in the same location as the adventurer, remove it from the room's possessions and place it into the character's.
DROP	D	Reverse of Take.
PRAY or CAST (spell)	P/C	Activates magic inherent in object, area, or character.
LOOK	L	Repeats description of area.
USE	U	May activate special feature of an item.
FIGHT	F	Engages foe, if present, in combat.
INVENTORY	I	Lists items character has.
EXAMINE	X	Takes a closer look at something.

Compass points (N, E, S, W, etc.)		Moves adventurer from one location to another if a connection exists in a specified direction.
QUIT	Q	Terminates game. If adventure is to be continued, may save data into a file.

The abbreviations are used in systems where string-handling is difficult or limited, or where a single character is needed, like in a realtime game.

The hard type is enclosed in the program; for example, typing "FIGHT" causes it to branch to a certain location or call a subroutine. In some areas this is wasteful, as in a room where there is nothing to fight. The soft method offers many advantages; however, it is restrictive in some ways. Included in the latest description of an area, together with what the character sees and a list of items which are within reach, is a list of actions that the character can perform. Inputting the number or letter of the choice causes an event to take place; for example, "PRINT 'There is a bright flash of light. Your sword is glowing.'" and "ADD 1 to SWORD-ATTACK." Still another method is to combine the types, have options that are standard, as well as some that vary from room to room. (For a better explanation of this principle, see issue #5 of *Sorcerer's Apprentice* magazine.) The two can be combined in yet another way: write (in a file) several lists of pseudo-commands that are executed whenever the specific command is typed. For example, 'TAKE' would ask for the item to take, check the contents in the room, and either add it to the character's possessions if possible or print an error message if it was not. When a character's strength is increased, the "weight limit" might be increased as well. With this method, commands could be more easily enabled, disabled, and changed; FIGHT would only mean something where there was something to fight. Items with special abilities could be "switched off" or even changed completely in certain situations.



With the latter method, commands can be deleted and added within the framework of the program. For example, if you destroy an idol in a certain room, it is only logical that all references to that idol be removed. You might write an "erase option" pseudocommand to include in the command sequence for options like these. Perhaps the easiest way to do the description part of this is to put in in "phrases"—that is, units that can be replaced as the need arises. In a particular room, there may be 6 of these:

1. "This is a large sized room "
2. "with a large, jade idol in one corner and "
3. "with exits to the "
4. "east, "
5. "west, and south. "
6. "A ladder also leads up. "

Thus, is one were to take the idol, phrase 2 would be deleted. Smashing it would cause the phrase to be replaced by "with a pile of green fragments in one corner and..." It also seems that you can take the ladder, replacing phrase 6 with "There is a hole in the ceiling, out of your reach." And, of course, the east door is lockable from the other side; doing this will change phrase 4 to "east (although this door is locked)."



Null strings are ignored. With this system you must be careful concerning punctuation and grammar; deleting "east" from "Doors lead east and south" produces "Doors lead and south." If you have the inclination, you might replace "door" with "doors" depending on how many exits there are. What is more important is a "formatter," a routine to make sure that a word is not broken at the end of a line and continued on the next line. One way to do this is to take a dummy string as many characters wide as your screen or printer and put words into it until it overflows, then backtracking one word and printing the string. With this it is even easy to "justify" the string (meaning to insert blanks between words until the string is exactly as wide as the output device, so that the left and right edges are straight instead of jagged).



Getting back to the hard command method, unless you wish to have the player type one word at a time, a "parser" is necessary. One of the more basic types can take two words and act upon them, as long as the first is a verb and the second a noun ("TAKE TREASURE," "FIGHT DWARF"). A more complex type, namely the one in Zork, first scans the input string to see if it knows all the words, then separates the sentence into clauses (separated by the word "and"). Each of these clauses has one noun (the object) which has an action done to it and possibly one noun (the actor) which does the action, as well as a verb (the action itself). The matter is complicated with adjectives and modifiers; the result is that the parser can understand such statements as "PUT THE RED BOOK ON THE TABLE." Here the actor is the character. "TABLE" is an item that can possess items, if they are given to it in the right way. "BOOK" is the object of the action, and "RED" is an adjective that is referenced in case the character has more than one book. All "THEs" are ignored. "PUT," of course, is the action, which removes the red book from the character's possessions and puts it into the table's possessions. This principle dates back to T. Winograd's work in robot control and *SHRDLU*. (See *Understanding Natural Language* and the July-August issue of this magazine for more information.)

A problem, similar to the "Charm Person" example, now rears its ugly head in many a would-be program—Monsters (the term refers to anything hostile in the dungeon). To keep them from being mere faceless attack and defense values, special abilities are often included. To cite a well-known example, werewolves may only be killed with silver weaponry. One way to implement this is to have a Boolean array called "POWERS," in which POWER(1) would be the ability to take nonvital damage without dying. This soon grows out of proportion: let's say you had twenty special abilities (actually a small number... the imaginative Game Master usually has hundreds if not thousands) and twenty monsters (again, a piddling value). Since a byte is the smallest accessible unit of memory, the table of monster powers is 400 bytes large. This number grows quickly as monsters or powers are added, and may well outdo your computer's memory sooner than you wish.

A pointer structure is again the answer. Each monster can have four powers from a list of hundreds. This is a realistic limit, but is expandable in case your monsters have more than four powers apiece. Each monster's POWER array is four unique integer variables that correspond to the master list of abilities. Since an integer is usually 2 bytes long, this type of structure can handle 50 monsters in the same space as the previous example. Also, this format is as efficient as the other, instead of saying

```
IF POWER (1) IS TRUE THEN do actions associated with it
IF POWER (2) IS TRUE THEN do actions associated with it
```

```

*
*
etc.
```

a simple loop might be necessary, as

```
FOR I = 1 TO 4 DO
  IF POWER (I) = 1 THEN do actions associated with it
  IF POWER (I) = 2 THEN do actions associated with it
```

```

*
*
etc.
```

This might be the solution to putting spells and other magic in. Of course, the main problem—that of "would this item,

spell, etc. work in this situation?"—still holds. At present it seems that each case must be decided individually.

Once you have both a good system and a good program for using it, you will need a good adventure to implement. This, of course, is up to you and your skills as a writer and designer. However, there are some general pointers that are useful:

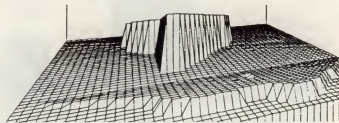
First, have some goal. A good one is getting out from a dungeon that one is trapped in. More common is the accumulation of gold or experience to a certain level.



Second, have subsidiary adventures along the way. These could very well yield helpful information or items.

Third, design it to be run from scratch. Not everyone may know that dimethyl ascorbate must be oxidized before it is edible, and if you have some in your adventure, many people will miss out on it unless you include a clue somewhere.

Peripherals and special features are tempting to include in a game design. Even with a joystick one can do a lot. However, remember that not everyone has a joystick—or music board, voder, or whatever. Games designed for the mass market should not contain these features. Random-access files, though, are fair game as disk drives are becoming so commonplace.



Still, they are fun to play around with. The value of some such as vocals (both for input and output) speak for themselves. (Sorry.) "Taps" played when a character dies is also hard to resist. Graphics are a perennial favorite, but the state of the art (without getting into the hundreds of kilodollars) is still not of photographic quality, which must be achieved before one can have a truly graphics-oriented fantasy game. Still, what is available is sufficient to draw inventory items and other helpful stuff. And with a bit pad or a touch-sensitive screen, you can not only manipulate items by hand but also explore on a map that is drawn on the screen or tablet. The example springs to mind of a corridor that has a trap in a certain section that must be circumnavigated—or else. Or how about a game where one could wield a light pen like a broadsword? The possibilities are limited only by how far your wallet happens to stretch!

After the adventure is complete, check it. It is a piece of software just like the usual type and is just as prone to errors (if not more so). You might find that the corridor that was supposed to lead to Room #35, the Ogre room, actually goes to #53, the Treasure vault. Have a friend check it out, and watch his game.



Incidentally, if I haven't hinted at it before, make your programs "user-writable." People certainly enjoy playing a good dungeon... but think of how much more fun they would have if there are hundreds of additional adventures (different in scope, yet bound under the same set of rules) floating around, some of which were composed by friends (or enemies)? Of course, you must remember the rule that large software firms seem to have forgotten: make sure it works before releasing it. If people are going to be plugging data into a program, make sure they can't plug the wrong data in, or at least tell them when they do. "Common sense," you say? You'd be surprised.

And a final, stern word of advice (although this article might already contain too many): before you hand it out or sell it or whatever, ask your self "could I make it better?" as a totally unrelated example, television reception would be ten times what it is today if industry had standardized not the first system that came along, but the best.

Enough reading. Get to work... you've got a universe to build. It's a big job, but somebody has to do it!

BIBLIOGRAPHY

The following fantasy role playing games are all recommended:

Tunnels & Trolls and *Sorcerer's Apprentice* magazine, available from Flying Buffalo Inc., PO Box 1467, Scottsdale, AZ 85252
Traveller and *The Journal of the Traveller's Aid Society*, available from Game Designer's Workshop, Box 432, Normal, IL 61761
In The Labyrinth and *The Space Gamer* magazine, available from Metagaming, Box 15346, Austin, TX 78761
Dungeons & Dragons and *The Dragon* magazine, available from TSR, PO Box 756, Lake Geneva, WI 53147
Runequest and *Wyrm's Footnotes* magazine, available from The Chaosium, Box 6302, Albany, CA 94706

The following books may also be helpful:

The Complete Strategist by J. Williams
 "Analyzing English syntax with a pattern-learning parser" by K. McConlogue and R.F. Simmons, in *Communications of the Assoc. for Computing Machinery*, vol. 8, no. 11
Understanding Natural Language by T. Winograd
Introduction to Artificial Intelligence by P. Jackson

In addition, the author would like to discuss problems and discoveries in the computer FRP games field, as well as games in general. Write to LPO Box 12613, Livingston C. Rutgers U., New Brunswick, NJ 09803.

PCNET

The time is 8 PM. In Los Angeles, Dick enters a message for his friend Jane in New York City onto his PAN system. To get low night phone rates, Dick "time lags" the message for transmission at 3 AM.



3AM: Dick in Los Angeles and Jane in New York City are both asleep. Dick's PAN dials Jane's number, verifies that Jane's PAN is ready, and transmits the message. Both PANs then hang up the phones. Dick's PAN shows the message as "sent." Jane's PAN reviews the message for review the next day on Jane's cassette tape.

PCNET, a project of People's Computer Company, has available a computer mail support software for the Commodore PET. Other versions for the Apple will be available soon.

cassette tape for use with an 8K or larger PET. All that is required is a telephone line, an auto dial-auto answer modem, and a personal computer (available at present only for the Commodore PET).

\$12; a user's manual is available separately for \$2.

The new PCNET computer mail system is called PAN—a program on

The PAN software, a perpetual license for its use and a user's manual sell for

If you would like more information on the PCNET project, or would like to order the PAN software including a perpetual license agreement, contact PEOPLE'S COMPUTER COMPANY, 1263 El Camino Real, Post Office Box E, Menlo Park, California 94025.

STAR TREK— a dialogue approach

part 1—background

BY SERG KOREM

"Space, the final frontier . . ."

INTRODUCTION

"... Let's see now, was fire torpedoes number 3? Or was that sensor scan?" How often have you played a Star Trek program and asked similar questions? Are you tired of fighting Klingons by "remote control"? Are you tired of fighting Klingons? Did the fun and thrill of playing that new Star Trek program warp away once you learned one game was pretty much like another? Have you become complacent with cruising through space for that matter? Or antimatter? If so, this series of articles may not only restore your initial excitement but hopefully it may teach you some simulation basics as well.

OBJECT

The intent of this series will be to lay out guidelines on how to build a Star Trek program "universe" which will recognize "conversational" as opposed to "numeric" commands, respond conversationally where appropriate, maintain a high level of interest and user interaction, and work on the smallest to largest of systems, single or multi-processor. The object of the program will be to achieve an output similar in form to the following:

KIRK: SCOTTY, GIVE ME MORE POWER IN THE PHASERS.
SCOTT: AYE, CAPTAIN. FORWARD PHASERS NOW AT 151, AND REAR PHASERS AT 145 UNITS.
KIRK: LOCK REAR PHASERS ON TARGET, MR. SULU.
SULU: SETTING, SIR?
KIRK: SETTING 85, MR. SULU.
SULU: AYE, SIR. SETTING 85.
KIRK: FIRE!
SPOCK: HIT ON ROMULAN AT 2.6. SENSORS SHOW HE IS NOW AT 30.012 PERCENT CAPABILITY. ROMULAN AT 3.4 AT 15 PERCENT POWER.
CHEKOV: ROMULAN AT 6.2 DESTROYED, SIR.
SPOCK: CAPTAIN, 5 ROMULANS JUST DROPPED THEIR CLOAKING DEVICE.

(This was taken from an earlier version of my program.) When I say "conversational," I mean grammatically correct, with the ability to be readily adaptable to speech synthesizers such as the Computalker.

APPROACH

Each article will deal with one aspect of the program, i.e., the ship, navigation, universe, etc. This should result in a highly modular program structure which will allow the individual programmer to choose which procedures he wishes to implement. A modular structure encourages expansion.

Although specific blocks of code will upon occasion be used as examples, most of the program will be laid out in a structured, non-dedicated language. Not only will this introduce the reader to a useful means of laying out program logic, it will also allow the user to implement the program in the language of his choice. Most of the specific code examples will be in BASIC although upon occasion I'll compare it to another language to point out BASIC strengths and weaknesses.

Even though I have written a similar program with great success, this will in essence be a totally new game. As such there may be, and probably are, ways of doing some things quicker, more efficiently, etc. Feel free to write to me (26 Marvin Dr. A4, Newark, DE 19713) to correct, suggest, or complain. Any suggestions used will be duly acknowledged in future articles.

A word about definitions. When I say

Procedure — I mean a block of code which processes data dealing with a particular function—navigation, sensors, etc. This is similar to BASIC GOSUB/RETURN pairs, and FORTRAN SUBROUTINES, but is closer in concept to ALGOL/TAL PROCs. I may also refer to this as a module.
Subprocedure — a block of code which gets repeated within a procedure. Nothing like a subprocedure exists in BASIC, but the concept is useful nonetheless.

Block — a group of coded statements.



Printed with permission from Paramount Pictures Corporation, "Star Trek—The Motion Picture."

THE PSEUDO-LANGUAGE

This is a tool used for defining logic flow without the need for flow charts. A program is first defined in this language and can then be coded in any programming language, APL, BASIC, etc. The first statement type is fairly straightforward:

Declaratives — any operation:

```
A = A + 1
B = SIN(A/2)
GOSUB 2300
RETURN
etc.
```

The other statement types are conditionals:

```
IF (test 1) THEN (block 1) ELSE
IF (test 2) THEN (block 2) ELSE
...
IF (test n) THEN (block n) ELSE
(block for else);
```

Note that all pseudo-language statements are terminated in semicolons. The quantities within () are not program code, but English language descriptions. The IF THEN ELSE conditional tests (test "n") if this evaluates true, (block "n") is executed. If (test "n") evaluates false, then the next IF THEN pair is executed. The final (block for else) is executed if none of the prior IFs evaluate true.

It is legal to omit (block "n") and/or (block for else) if no action is desired. So

```
IF (x is greater than y) THEN ELSE
IF (x is less than y) THEN ELSE
(print x);
```

would output the value of "x" when $x = y$. Note that this is equivalent to IF (x is equal to y) THEN (print x);.

The statement

```
REPEAT (block) UNTIL (test);
```

will execute (block) until (test) evaluates true. It is implicit in the structure of the statement that (test) is evaluated at least once and must reset (test) to false. On the other hand,

```
WHILE (test) DO (block) END;
```

will execute (block) as long as test evaluates true; that is, (test) is evaluated first.

```
FOR (range) DO (block) END;
```

will execute (block as long as (range) some value is within the range. (range) is usually specified as 1 = 1 TO 3 BY 1 or similar statements. This is merely a modified BASIC FOR statement.

(block)

is defined as being;

```
BEGIN (group of statements) END;
```

where (group of statements) can be a nested pseudo-language statement set.

!Comments are denoted by imbedding them with exclamation points!

This is all that is needed to structure a program. Readers may have noticed that there is no GOTO statement. Although they may be used, virtually all programs can be structured (although not necessarily coded) without them. In this manner program flow will tend to be top-down and will result in more readable code.

INTERACTIONS

The first step in setting up a simulation is to define all of the procedures and how they will interact. This can be done in several ways—flow charts, descriptions, or block diagrams. Because we are dealing with general concepts at this point, we will use block diagrams followed by a brief description of each element. Figure 1 is the Interactions Block Diagram.

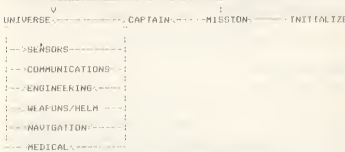


Figure 1 Interactions Block Diagram

INITIALIZE — Define and set all program variables. This includes randomizing the random number generator.

MISSION — Determines the goal of the game. This can be as simple as "destory 76 Klingons in 32 days" or as difficult as hand to hand combat with a Gorn.

CAPTAIN — This is the central unit of the game and the person who will react to the situations created by the program — the use. Although this can be hardcoded as James T. Kirk (the captain on the show), it is usually the actual name of the player. More on this later.

THE SHIP — THE U.S.S. ENTERPRISE or whatever you prefer. **SENSORS** — Instruments which provide information about the universe to everyone aboard ship. **COMMUNICATIONS** — Allows the Enterprise to communicate with various and sundry ships, entities, etc. in the universe. The sensors and communications are the only sources of information about the universe the Captain has at his disposal, except for God or any other powerful being or force capable of bypassing the ship's systems. **ENGINEERING** — Maintains the ship's systems. Sort of a ship's utility company. **WEAPONS/HELM** — Just that. Allow the ship to defend or attack and maneuver within the universe. **NAVIGATION** — Tells the ship how to get "there" from "here." **MEDICAL** — Will play a minor role in this program, but can be expanded if desired.

UNIVERSE — All else. Other ships, starbases, planets, suns, life forms, etc.

DIALOGUE

Since this is to be a dialogue-based game it would be perfectly valid to set up a procedure to handle this, but because the series is geared to systems of every size, it is better to dynamically generate conversions as needed in order to

Continued on pg. 26

MARKETPLACE

Classified ad space available: \$60 per vertical inch.
Columns are 3 1/4" wide.

COMPUTER SUB HUNT

Track an enemy sub around an unknown ocean, get close and fire depth charges. Send \$7.95 plus \$1.00 shipping to SATALITE COMPUTER CO., 38 Bourne Avenue, Tiverton, RI 02878—order now—For the TRS-80—Level I.

ComputerTown, USA!

A grass roots, economical model of how to offer everyone in a community of 27,000 the opportunity to use a microcomputer. Your interest and help is invited. For more information, please send a large self-addressed, stamped envelope to: ComputerTown, USA!, Box E, Menlo Park, CA 94025.

PCNET PAN

An electronic mail package, PAN allows PET owners to send and receive messages over the telephone network. Entirely written in BASIC, PAN permits immediate message transmission, or unattended transmission at a specified time. PEOPLE'S COMPUTER COMPANY, PCNET Project, P. O. Box E, Menlo Park, CA 94025.

for TRS 80 II. Compucolor II.
& PET

EVERYTHING YOU NEED TO BRING YOUR BASIC PROGRAMS TO LIFE WITH SOUND!



SOUNDWARE is a complete package—YOU GET A SPEAKER/AMPLIFIER UNIT complete with connectors. No wiring or soldering. Just plug in!

YOU GET A DEMO PROGRAM with a variety of sample sound effects—sirens, laser sounds, tunes!

YOU GET A COMPOSER PROGRAM to help you create your own original sound effects. Fun for all ages!

YOU GET AN INSTRUCTION BOOKLET that tell you how to insert sound into your programs. 1 year warranty.

Complete Package or Booklet	
ALL PETS & TRS-80 II Compucolor II	\$29.95 \$39.95
Programs	\$14.95 \$19.95

SEND FOR FREE CATALOG OF GAMES FOR PET & COMPCOLOR!!

SEE YOUR DEALER TODAY! Or order direct from CAP by phone or mail. VISA & MasterCard orders include expiration date. Add \$1 postage & handling per order. \$3 for air or COD. Arizona residents add tax.

CAP Electronics 8462 Hillwood Ln., Suite B, Tucson, AZ 85715 (602) 296-4978

WILEY SELF-TEACHING GUIDES... Keeping up with the fast-moving world of recreational computing

From the Editors of *Recreational Computing*...

TRS-80 BASIC

Bob Albrecht, Don Inman, & Ramon Zamora

Packed with games, graphics, and practical applications, this eagerly awaited guide leads you step by step to maximum use and enjoyment of your new TRS-80.

"The book is OUTSTANDING and is perfect for all LEVEL II owners... geared toward the beginner, but does contain a wealth of information for more advanced programmers... TRS-80 BASIC is the best thing to happen to LEVEL II owners!"

Computronics Monthly News Magazine

0 471 06466-1 1980
351 pp. \$8.95 paper

STRUCTURED COBOL

Ruth Ashley

Produce programs that are easier to read... less costly to debug. With this complete self-instructional guide to the computer language most widely used in business, you'll write or modify programs more swiftly and accurately than you ever thought possible.

0 471 05362-7 1980
295 pp. \$8.95 paper

JOHN WILEY & SONS, Inc.

605 Third Avenue
New York, N.Y. 10016
In Canada: 22 Worcester Road, Rexdale, Ontario
Prices subject to change without notice. 1-7715



BACKGROUND MATH FOR A COMPUTER WORLD, 2nd Ed.

Ruth Ashley in consultation with Nancy B. Stern

Here are all the basic mathematical techniques, facts, and concepts you need for productive, intelligent interaction with your computer using any computer language. The new edition of this popular manual includes an entirely new chapter on trigonometry.

0 471 08086-1 1980 304 pp. \$7.95 paper

These are the newest titles in Wiley's fast-growing line of popular STGs to help you make the most of your personal computer. Look for all of them at your favorite bookstore or computer store!



ON FUTURE FANTASY GAMES

BY DENNIS ALLISON

Introduction

Even the most complex of today's computer fantasy simulation games is simple-minded compared with what is possible. Limitations in concept and execution appear in many places. Most games are built around the simple "explore and collect" scenario. Interaction with the fantasy environment is limited by the computer hardware on the one hand and by the limitations of the software on the other.

Pictures And Sound Would Help

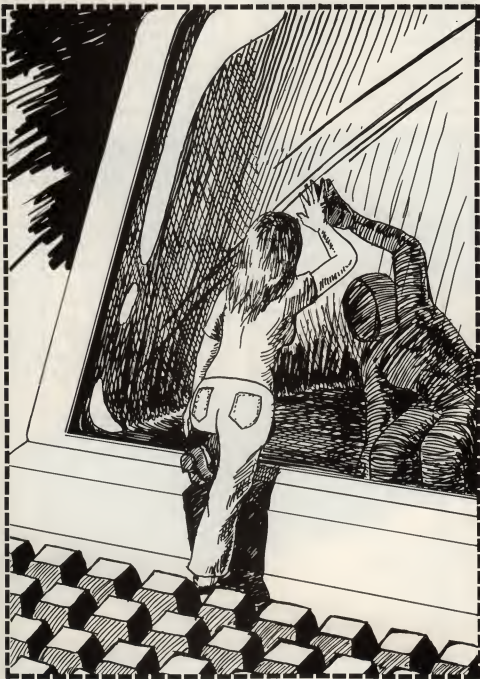
We humans depend upon our eyes and ears; they are our primary input devices. When the game tells us what to see we create the vision with our mind's eye. Literary devices aside, if we could see and hear what was happening the game would have the immediacy of a movie or TV program and less that of a book.

One thinks immediately of animation, drawing the events of the game out onto the computer terminal's screen. Some games make limited use of the computer's ability to present information graphically (maps, for example) but none provides true animation. The necessary resolution for an acceptable quality picture is not available nor is the processor power. It takes a lot of computation to make a single minute of animation.

Video tape might appear to be a solution. A control unit is available from Cavi Systems (26 Trumbull Street, New Haven, CT) which allows a computer to control a videotape machine. But tape is a linear medium; you can't switch around as a game demands.

Enter the video disk. It seems tailor-made for the gamesmith. Video disks have been designed primarily as movie delivery systems and high quality sound systems. But range of application will be much larger: digital data storage, computer-assisted instruction, and making computer fantasy games more real, to mention a but a few. U.S. Pioneer Electronics (85 Oxford Drive, Moonachie, NJ 07074) has recently begun marketing the video disk player shown in Figure 1 for a suggested retail price of \$750. It uses a recording technology developed by Phillips N. V. in Holland over the last ten years. While it is not designed for computer control, it is almost certain to be adapted for such use.

The primary advantage of a video disk is its ability to access any frame or sequence



of frames randomly. A standard disk can store 54,000 frames, each of which holds one screen picture. In normal sequential mode, that's enough to hold one half-hour segment of a movie. Digitally that's a lot of information, over 10,000,000, 000 bits or some 1200 floppy disks. Frames located close together can be referenced with only minor interference with the video signal; the whole space from frame zero to frame 54,000 can be traversed in under 15 seconds.

The recording technology uses tiny pits in a reflective surface such as those shown schematically in Figure 2. The disk is scanned with a laser and the level of reflected light is measured to reconstruct the signal. The receiver mechanism is shown schematically in Figure 3.

Since the playback mechanism never contacts the disk, a single frame can be displayed continuously without any damage to the disk itself. Disks are relatively inexpensive to manufacture in quantity though the mastering process is complex; prices delivered to the consumer will be in the \$10 to \$20 range.

Using A Video Disk In Fantasy Games

The idea is obvious: use the video disk to add pictures and sound to the game. That means careful planning because even a half-hour of video is short in a complex game; many scenes should be of the single frame variety.

One would like to be able to use animation and written output in conjunction with the output from the video disk. The video disk conforms to standard NTSC format for television signals. Unfortunately, most computer systems do not conform to that standard format because of a desire to produce higher resolution outputs. Of contemporary computers only the TI 99/4 and the Cromemco SDI board are truly NTSC compatible. The Cromemco board actually has the mixing facility on-board already.

Computer fantasy games are not the application for this mixture of computer output and recorded picture and sound. Already the same idea is finding application in computer-assisted instruction. An intelligent video disk program titled "Diagnostic Challenges" will premiere at the XI International Congress of Gastroenterology in Hamburg, West Germany in early June. Produced by Smith Kline and French Laboratories and controlled by a Western Digital Micro-engine it will provide for user interaction through a specially designed keyboard.



Figure 1. The U. S. Pioneer Videodisk provides random access to 54,000 frames of video information.

VIDEODISC CONSTRUCTION

MAGNIFIED VIEW OF A VIDEODISC

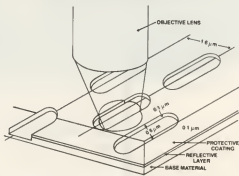


Figure 2. The recording technique utilizes small pits in a reflective layer on a plastic substrate.

Sounds And Speech

Speech is a familiar and natural medium of communication. The terminal keyboard is not; in fact, many people who would like to play computer fantasy simulation games will not be able to type. This suggests other means of input would be desirable including touch screens and, most importantly, speech.

Speech recognition is a difficult problem. Several systems are on the market with single word or phrase recognition vocabularies of a few tens or hundreds of words after training. Figure 5 shows the Speechlab system for S-100 based systems produced by Heuristics (1285 Hammerwood Avenue, Sunnyvale, CA 94086). Much the same recognition process is used by all speech recognition systems. The audio signal is sampled, normalized in time, and passed through filters to develop a signature characteristic of word sound patterns. The observed pattern is then matched against a stored dictionary of patterns to find a "best" match, which is then reported as the word recognized. These systems work well only with isolated words or phrases but fail for continuous speech. Fortunately, speech input for a game can be trained and can use a small vocabulary. But the constraints are irritating.

Speech output can come from the video disk (there are two channels) provided its content can be predetermined. Alternatively, it can be synthesized directly by the computer. Limited vocabulary speech synthesis devices are beginning to appear; the Texas Instruments *Speak and Spell* is a good example. Sounds other than speech can be synthesized by special hardware developed to support the video game industry. Alternatively, one can use a microprocessor to manipulate a speaker in the fashion of the Apple II.

Appealing To The Other Senses

There are five senses. We neglect the other three: smell, taste, and touch.

In the 1950's someone proposed "smell-o-vision" where odors could be prepared to fit the displayed action. Caves would smell dank and wet; monsters would have stomach-retching breath. It would have a limited market perhaps, but such a peripheral could be made easily; it would fit just above the terminal and would have to include a small fan to keep the last scenario's smells from intruding upon the present action.

One could imagine a headgear mechanism which would squirt into one's mouth the appropriate flavoring agent to match the

PLAYBACK LASER PATH

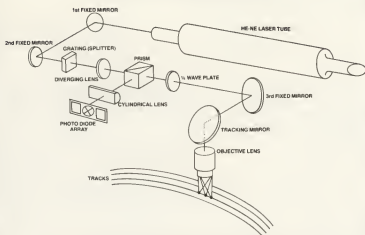


Figure 3. A laser is used to scan the disk and measure the reflectance to reconstruct the recorded video signal.

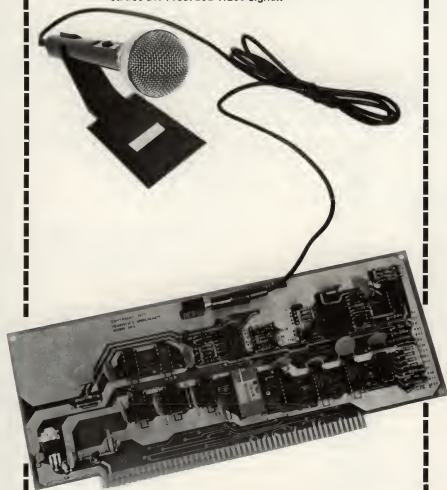


Figure 4. This Heuristics Speechlab system can recognize up to 250 different words after appropriate training.

story. A trip to *Candyland* might be quite an experience.

And then there's the possibility of feelies where the tactile senses are exercised. The computer would provide touch responses—the caress of a lover, the blow of a sword—all in response to the current scenario. Given the violence many adventure games seem to have, an effective implementation might actually be dangerous.

Other Plots Are Possible

Even without all the fancy hardware, the limitations of today's fantasy games are primarily those of imagination and computation power.

The exploration paradigm has been popular because it's easy to implement and understand. But the imperative fast-moving adventure is only one of a universe of possibilities. In most games the interaction, albeit complicated, is pretty superficial. Games where the interaction is even more complex could be even more fascinating. Some different approaches were sketched in my *Epic Games* article published in the March-April 1978 issue of *People's Computers*. Consider, for example, the electric murder mystery. You, the player, might be cast in the role of the detective. Through the magic of video you might visit the scene of the crime, interview witnesses, collect evidence, and interact with a complex world. The plot and characters need not be fixed but could be developed from the program using built-in rules about people's motives, the physical act of committing a murder, and the myriad of other facts which go into developing a plausible plot for a mystery story.

Another kind of interaction might be the electric biography. Here the player would be able to interview the person and ask questions. Here the knowledge base would need to know about the personalities and events which shaped the person's life. Using this knowledge base the computer could select scenes and construct answers to the player's questions. Imagine being able to discuss palace politics with some group of courtiers from the court of Louis XIV!

Games of this complexity are not yet possible. But they are not too far from what is possible.

Many of the ideas here are supportable with techniques borrowed from artificial intelligence. The only real demands are large amounts of memory, large mass storage (the video disk might help here) and major quantities of computation power.

CRYPTARITHMS

BY JOHN DAVENPORT CREHORE

Send your solutions to Jack Crehore,
P.O. Box 96, Charlotte Court House,
Virginia 23923

Here's a thrill! It is thrilling when you succeed in ferreting *arithmetic* Sequences out of code letters! The fun...uh-h...toil that I've had so far with Puzzle 27 persuades me that other near-Geniuses might enjoy the bit of educational exercise it offers.

I haven't solved it yet. I'm copying from my scratch papers, "6-6-41 Good problem; too hard for beginners."

So long as we're having fun, listen, you who keep urging me to use real words for keys instead of the 14 letters I now use. Real words give unfair opportunity to anagrammers to send in solutions without having done any mathematics at all.

Anyway Puzzle 27 (Genius), before I transposed it from your alphabet lingo into my 14 cryptic letters, Looked kind of cute to me at my advanced age of 89! I must have thought young in 1941, when I used the real alphabet. Here's 27: FOAMY DRINK and 8 little words under it in a subtraction example. Top line OKIFMANDRY. If I gave you more, you'd solve it.

HINTS: Start with *Sequences* found in Puzzle 27. The general procedure is simple. List every group of letters (digits) in which you can indicate actual differences in value or comparative differences in size—large or small.

Pick a letter, let's say C, and make a list of letters you have determined may be greater than C in digit value. When you find four digits larger than C, you know C can't be larger than number 5. This way you often can find values for several letters close enough to break an impasse. I find without my 1941 scratch:

H > C Col. 10
A > C Col. 4
R > C Col. 7
Y > C Col. 7

OK. That's my share! Rub your lamp and call your Genie to help you finish Puzzle 27. You'll need him!

HINT: In an Addition example if a column has three digits (addends) and one of them appears as the digit of the Sum (total), then the sum of the other two digits *must* be 10.

H	4
F	6
K	7
RK 17	

Note also that this always gives a carry to the next column of just 1 because with 10 for two letters, the third letter can't bring the total higher than 19. Note also that unless both letters are the same, neither letter can be 5. Besides, when you solve one letter you solve the other to make 10 total.

OK so far! But with a longer column, say 5 or more digits tall, it will still often be true that if one letter of the total (the Sum) appears in the column above it, then all the other letters add to a total of 10, 20, 30 or whatever.

Look back! I made up this situation in Puzzles 19 and 20, in RC 44 Mar-Apr 80. Under Puzzle 20 (Computer) I gave the Hints: "Simple arithmetic addition example. There is one column that will break the code, without a computer. NINE HEX" (Col. 4)

HINT: When you're stuck amidst a hard Puzzle, copy it off and fill in the solved digits close to their letters. Then test values at random for all nearby letters.

CORRESPONDENCE

S. R. McENTEE: Beautiful work—so clean. Hard Puzzles don't fool you, I see. Thanks.

JUDY: { Your letters came to my desk stapled together. Keen solvers, having computers and not using them!

Judy is our first girl! I hope Hobo is another, our second. Thanks for: "More! More!" and "Enjoyed the puzzles!"

BOB: Headwork and computer too! You think fast if 20 minutes for all

that scratch was *clock* time. I can see that common puzzles wouldn't throw you.

BOBBY: Clever! Thinking up an Elimination Table! The use of one was taught to me in 1942 at a national convention of the American Cryptogram Association and The National Puzzlers' League. My name? John, yes; familiarly, Jack; NINE HEX in several of my statuses. "Crehore" toned down from feudal Irish "Crohoore" found in Plymouth colony in 1641.

RICHARD: Yes, two solutions to old No. 7. Everyone called it the hardest, but they solved it! I gave useful clues in columns 1, 3, 4, and 7, and fixed Y as digit 1, or 2, or 3. I liked it all too well to leave it unused even though I couldn't get rid of the dichotomy in B and R. I'd like clear record of your name; can't decipher two letters of your surname. Write again!

K8VDU Fred: Thnx for the wallpaper, Good Buddy. With a land line up we could modulate, eh? Come back! So you are "Fritz" in A.C.A.! Like you, I've been in both A.C.A. and N.P.L. First joined, 1936; after lapses just rejoined A.C.A.

I never received your solutions to Nos. 9 to 12. I'll send you xerox for only S. A. S. E, and list your score! Puzzles 10 and 12 were mutilated. Do solve No. 11 for me! It is a *triumph* of mine! It is simple, logical and taintless. Still, not one of our RC fans, Geniuses, has yet claimed to have solved it without a tip! It's Cancellation for 5th grade kids. Seventy-threes. Eights. NINE HEX.

CHARLES: Your Computer Program never reached me. Anyway, I'm only slightly qualified to attend to computer affairs. I quit giving word keys years ago. They make invalid any certainty that a contestant's solving be solely mathematical. I promote these fourteen letters—ABCFHJKMNRUWY. They stand differentiated even in careless script. Do write again. Programmers must rank as experts! And we're all in this to learn, eh?

Jack Crehore—NINE HEX

Puzzle 25 (NOVICE)

D V D
V V D
D V D
N V N D

SINIS: "This is one to broaden the mind..." caption in my file. Remember it's a PUZZLE you are working on. Don't start with fixed notions. There is no error.

Puzzle 26 (ADEPT)

T C F J N B
Y T A C
B B T Y M K
M T C J Y B A
C B K M A F
N C C T Y N
F J N T C B J K K

...!
is detected at a glance. After that routinely tested due to position range of three small letters that are in this Issue 46, applies here. The for an Adept, in COMMENTS HINTS: One of the Hints, routine

PROBLEM 27 (GENIUS)

H B J C R F A Y K M
- C H K R M F J Y H B
J B Y A B C B M M

HINTS: A study in Sequences. I am sequencing clues amongst COMMENTS so that experts, by unintentional glimpses, do not lose the satisfaction they might have had from solving solo. Look for remarks that concern Puzzle 27. Remember HINT in RC40, to chart values as small or large.

Cryptarithm Hints by NINE HEX

PROBLEM 28 (COMPUTER)

H U H H T R J
M W H W T J
Y W H W T W
M W H W T J
N W H H T J
B N H W H B J
M T H U R H N W

HINTS: There is no Carry-to Column 5. With this knowledge you can fix T and W!



The ATARI® Tutorial

IRIDIS

The IRIDIS #1 tutorial for the ATARI is available now! You get a C-30 cassette or a high-quality diskette with four excellent programs for your ATARI, ready to "Load" and "Run". You also receive the 32 page IRIDIS GUIDE which provides clear instructions for the programs. The GUIDE includes Novice Notes for the beginner, and Hacker's Delight for experienced programmers.

Our programs are written to be studied as well as used. The GUIDE will have complete source listings of selected IRIDIS programs. Not just listings, but an explanation of what's going on. If you are new to programming, IRIDIS is one of the easiest ways you can learn advanced techniques. If you're an old hand, you'll still find IRIDIS to be a rich source of ideas and ATARI techniques.

ATARI is a trademark of ATARI, Inc.

Please send me IRIDIS #1 for my ATARI immediately
☐ \$9.95 Cassette (needs 16K) ☐ \$12.95 Disk (needs 24K)

Name _____
Address _____
City/State/Zip _____
☐ VISA ☐ Card Number _____
☐ MasterCard Expires _____

Published By:

Dealer
Inquiries
Invited

The Code
Works™

Box 550
Goleta, CA 93017
805-967-0905

Programs for your ATARI®

PET COMPUTER TEACHERS

New! Everything you always wanted in arithmetic software. The PET PROFESSOR is a 70 program package which teaches with a moving cursor technique...add, subtract, multiply, divide whole numbers, fractions and decimals. \$299 on cassettes or \$259 on 5 1/4" diskettes. Send \$5 for a sample tape.

Our STUDENT WORKBOOKS for beginning programmers has classwork, homework, hands on and off exercises. FEED ME, I'M YOUR PET (beginner) and LOOKING GOOD WITH YOUR PET (intermediate) are \$4.95. TEACHERS' PET (lesson plans, quizzes, answer key) is \$4.00



COW BAY COMPUTING
BOX 515, MANHASSET, N.Y. 11030

THE



WONDERFUL WORLD OF EAMON

BY DONALD BROWN

The troll smirks at you as you raise your puny sword and prepare for battle. His mood quickly changes when he sees a bright flame appear around your blade. The battle lust is upon you once more; you attack!

If you have never experienced this type of excitement in your life, you have missed one of the newest types of games around. This is a scene from a Fantasy Role-Playing game (FRP for short).

Unlike most games, there is no clear single goal in an FRP. Instead, you are directing the actions of a character to achieve the goals you want him/her to achieve. Although most people accept normal goals such as 'make lots of money' or 'become a feared warrior,' you can choose any yardstick you want to measure your success. After all, it's your life—you ARE the character. (For a more complete explanation of Role-playing games, see the May-June 1979 issue of *Recreational Computing*, "What Is ALL This Stuff?").

The first major FRP was *Dungeons and Dragons*, created by Gary Gygax and Dave Arneson. Since then many other sets of rules have appeared on the market: *Runequest*, *Tunnels and Trolls*, *The Fantasy Trip*, and others. They all work in roughly the same way; a special player called the *Dungeonmaster* or *Referee* beforehand secretly designs a building or cave complex and places monsters, traps, and treasure inside of it. Then, the other players join in the game and they direct their characters through the dungeon, receiving descriptions of what they see and describing what they do. In conjunction with the rules, the *Dungeonmaster* determines who survives the inevitable combat and how much loot they bring home.

It has occurred to some people that at least the second part of *Dungeonmastering* (the interaction with the players) is ideal for a computer, and thus they set forth to adapt the games for their own systems. Unfortunately, troubles arose. In addition to piddling questions of copyright infringement, the simple fact is that these systems do not adapt well to computers.

The first major problem is in combat. In these games, when Grimy the Gnome decides to swing at an orc (or vice versa), the *Dungeonmaster* has to consider a large number of factors and calculate the chance of Grimy actually hurting the little beggar. Since humans by and large do not enjoy doing half a dozen major additions or multiplications, the games usually use large tables to calculate chances to hit. Unfortunately, while the computer has no objection whatsoever to heavy arithmetic, storing tables uses up definitely limited memory.

Even more of a problem, however, is magic. Since these games are set in places where strange things happen, the players want in on the power of magic, too. Thus, a wide set of spells are developed for the game. These range from the widely useful, such as throwing fireballs, to spells that are only effective against blue-eyed dragons. For all of their wonderful memories and lightning-fast computational abilities, computers are incredibly stupid and teaching them how to handle all of these spells is a frightening thought indeed.

Thus, something new was needed. So, with your kind permission, I present a new set of rules to be used freely by anyone who cares to: EAMON!

Eamon is a world at the center of another galaxy—instead of revolving around any star, all of the stars revolve around it! Those huge bodies exert strange effects on the world of Eamon, bending light, gravity, time, even the laws of nature themselves! The 'normal' world of Eamon is a rough, bruising, swords-and-sorcery society where the standard job is fighting Black Knights and dragons. However, at times reality shifts and you may find yourself facing Billy the Kid or Darth Vader!

Your character has three basic numbers that determine where his or her physical abilities lie. These attributes are all rolled on 3D8, and represent your Hardiness, Agility, and Charisma.

(A special format is used to describe random numbers. It is written xDy, and means that you generate 'x' random numbers from 1 to 'y', and sum them. Thus, 3D8 gives you a number from 3 to 24. This notation comes from the strange dice used in manually-run FRP games where there are indeed eight-sided dice.)

Your Hardiness is a combination of strength and constitution. It has two major effects: it determines how much damage your body can take before it dies, and it limits the weight you can carry. A player can carry up to ten times his/her weight in gronds. A grond is a measure of weight, which is also subdivided into ten dos. With the stellar tides varying weights so much, the ex-

act metric equivalent of a grond cannot be determined; however, an average weapon weighs about three gronds, and a coin weighs about a do. If your character has the maximum Hardiness of 24, he/she can carry a great deal, which makes a do about nothing.

Your Agility measures how quick your reflexes are, as well as how accurate. High Agility helps you in combat.

Your Charisma may be the most important attribute. Although it is strongly related to physical beauty, Charisma also includes how persuasive a talker you are, and other things. High Charisma will get you lower prices on your purchases, higher prices on your sales, and help you make friends among the denizens of the dungeons.

All three of your basic attributes will not normally change through adventuring, although special magic items may raise or lower them. Although the average value of your attributes should be around 16 or 17, the normal citizen of Eamon has Hardiness, Agility, and Charisma all equal to 10. This is to reflect the fact that only people who are a cut above others would normally go adventuring.

At least in the parts of Eamon that you will be exploring, Eamon is a rough and violent world. Thus, the mechanics of combat are very important.

Roughly speaking, all weapons can be divided into five classes: axe, bow, club, spear, and sword. These all attack in different fashions, and knowing how to use one type of weapon (such as a club) does not help you use another (such as a bow). Furthermore, some weapons are easier for the inexperienced person to use than others (the novice who picks up a club is more likely to hit his target than if he uses a bow and arrow).

Thus, we have the concept of weapon expertise. For each type of weapon, you will have a number that is added to your chance of hitting. These numbers are different for each type of weapon, and start out at the following values: Axe = 5%, bow = -10%, club = 20%, spear = 10%, and sword = 0% weapon expertise.

Your weapon expertise will rise as you learn how to use your weapons. Every time you successfully strike a blow, it may be possible to learn something new about the way to use that weapon. Since obviously all you can learn about is what keeps you from being perfect, your chance of learning from a blow is the same as your chance of missing. When you learn something, your weapon expertise in the class you are using will

go up by 2%.

The second factor is the quality of the weapon, called the 'weapon complexity'. A weapon that has been forged with good balance and of fine quality steel is more likely to hit than a cheaply made rush job. (Of course, it will cost more too!) Basic weapons can be bought with either 10% complexity (good quality), 0% (fair), or -10% (poor). Weapons with higher complexities still may be found in the caves and buildings around Eamon, and you may take them as soon as you kill their previous owners.

The third major factor that affects your chance of hitting is your agility. Twice your agility is added to your chance to hit. Except under the influence of magic, under which all things are possible, this will never change.

The last normal adjustment to your chance of a successful blow is based on the armor you are wearing. Armor will usually absorb some of the damage of blows that strike your body. However, there being no free lunch, you pay for this protection by lowering your chance to hit (due to the constricting effects of the armor that you are wearing).

There are three different types of armor you can wear—Leather, which will absorb one hit from every blow but lowers your chance to hit by 10%; Chain which absorbs two hits but lowers your chance of hitting by 20%; and Plate which will take five hits in every blow but lowers your chance of hitting by a full 60%. With any of these armors (or with no armor at all), you may also carry a shield which absorbs an additional hit per blow, while lowering your chance to hit by 5%.

These adjustments are for the player who has never worn any armor before in his/her life. Experience can train a person to compensate for these effects. Thus, each player also has an armor expertise, which starts at zero, and rises by 2% jumps just as weapon expertise does. However, your armor expertise can only cancel out the effects of armor—if you are wearing Leather armor and your armor expertise is 10%, any further attacks will not raise your expertise, unless you put on more armor. Furthermore, if you have built your armor expertise up to 20% while wearing chain armor and you move back to Leather, the net effect of armor will be 0, not increasing your chance to hit by 10%!

Once your total chance to hit has been determined, the computer will generate a number from 1 to 100. If the number is less than or equal to your chance to

hit, you got him! Now the question is, what did you do?

Both players and monsters (the generic term for non-player characters in Eamon—not all monsters are bad!) have a Hardiness figures, which is from 3 to 24 for humans. This number represents, among other things, the amount of damage the fellow can take before dying. On every successful blow, a random number is determined—this number varies according to the weapon and is expressed in NDN format. Any damage absorbed by armor is subtracted and if the result is greater than zero, it is taken off the target's Hardiness. This damage is not permanent, as it can be healed by either appropriate magic or rest at the player's home. However, if the player's Hardiness ever hits zero, he/she dies.

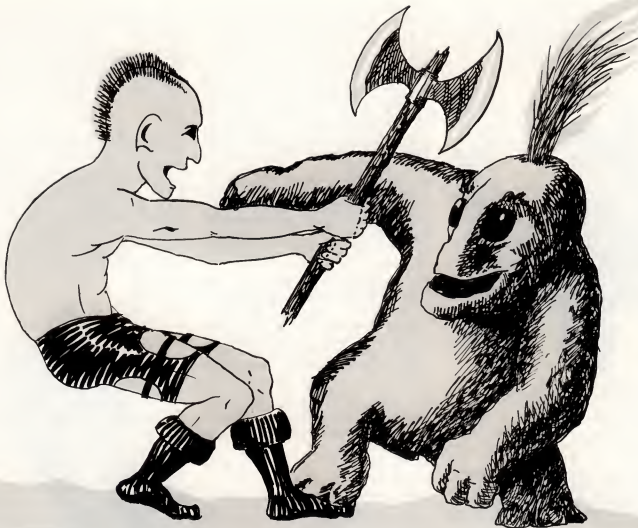
All of that, of course, is what usually happens. Sometimes, however, your weapon strikes a sensitive spot, making a critical hit. Sometimes your weapon strikes the ground, making a fumble.

Fumbles take place about 4% of the time, regardless of your chance to hit. What happens next is up to the gods: 35% of the time the fumble is recovered from with no bad effects; 40% of the time the weapon is dropped. (Monsters are subject to fumbles. If a monster is using his natural weapons, such as teeth or claws, and drops his weapon, he simply recovers.) The weapon being used is broken 20% of the time, with a 50% chance of hurting the one using it. The last 5% of the time, the user strikes himself.

Critical hits take place 5% of the time, again no matter what the chance to hit is. A critical hit will slip through your opponent's armor 50% of the time so that he receives all of the damage done (the armor does not absorb any hits this round). Otherwise, a critical hit does more than usual damage: three-halves of normal damage is done 35% of the time, twice normal 10% of the time, and three times normal 4% of the time. One percent of all critical hits have the effect of instantly killing the opponent.

To see all of this in action, hear now the tale of Hedric and the Troll.

Hedric was a starting character with a Hardiness of 18 and an Agility of 20. (Unfortunately, he had a Charisma of 4, but that's not relevant here.) Being a beginning character, he spent his starting gold to buy a good quality axe (complexity = 10%) which does 1D6 of damage. He also bought himself a suit



of leather armor. Thus, his chance to hit was 5 (weapon expertise) + 10 (weapon complexity) + 40 (twice his Agility) - 10% (the effect of armor), or 45%.

Hedric was strolling down a dark corridor when suddenly he was attacked by a fierce Troll. Although Hedric didn't know this, the troll had a Hardiness of 30. He used his fists to do 1D4 with a 40% chance to hit. The troll's rock-hard skin acted as a natural armor, taking 2 hits per blow. (All monsters have worn their armor long enough to have raised their armor expertise to the maximum; thus, no calculations for armor are done.)

It took Hedric no time at all to realize that he had to fight this thing, and the two began attacking. In the first round, the computer came up with a 23 for Hedric's battle roll, so he hit! Unfortunately, only one point of damage was done. The troll's armor took this with ease and the troll was unhurt. However, there was a 55% chance of axe expertise and armor expertise going up by 2%.

The computer generated random numbers of 12 and 54, so both attributes did indeed go up. Thus, on his second blow, Hedric's chance of hitting will be 49%.

Simultaneously, the troll attacked Hedric. However, his battle roll was 100—a fumble! The computer gives a random number, checks its table, and comes up with the result of a dropped weapon. Since the Troll cannot drop his fists (well, you know what I mean), he recovers without any problem.

The battle continues. Hedric makes most of his blows strike home, raising his armor expertise to 10% and his axe expertise to 8%. Thus, his new chance to hit is 58%. He has gotten 20 points through the troll's armor, leaving him with 10 points.

However, the troll has not been lagging either. Hedric has taken 15 blows, leaving him with only 3 points before death! He has also managed to raise his chance of hitting to 52% (monsters also learn from experience).

None of this, of course, does Hedric know. He has been told that the troll is 'very badly injured' and that he himself is 'at death's door, knocking loudly.' He has no inkling how much, if any, his armor and weapon expertise have risen.

On the next round, Hedric gets a critical hit! The result is doubling damage. Hedric does 11 points of damage, which after 2 are taken off for armor, leaves the troll with only 1 point. The computer checks to see if his axe expertise goes up (his armor expertise cannot, since it is at its maximum), and the troll gets his attempt.

Luckily for our hero, the troll fumbles again, this time breaking his weapons. As his fists disintegrate, one of the pieces hits him in the forehead, doing two points of damage. (After all, this is FANTASY role-playing.) This brings the troll's Hardiness down to -1, and he is dead.

Now that Hedric has defeated the troll,

he will have ready access to the troll's treasure. No matter what it is (almost), he will be able to sell it and use the gold pieces for something. One of the best uses is to learn a few magic spells.



On the whole, magic in Eamon is a very tricky thing. Due to the constantly shifting forces that power the strange happenings on Eamon (called magic for lack of a better word), a spell that always does a certain thing in one location may do something entirely different just two-hundred feet away! Thus, there are no Gandalf-like wizards with a thousand spells wandering around the countryside.

However, after years of careful experimenting, four spells have been developed that do work in almost every part of the normal Eamon world. (In the parallel worlds that one can fall into, the laws of magic are usually quite different, if magic exists at all!)

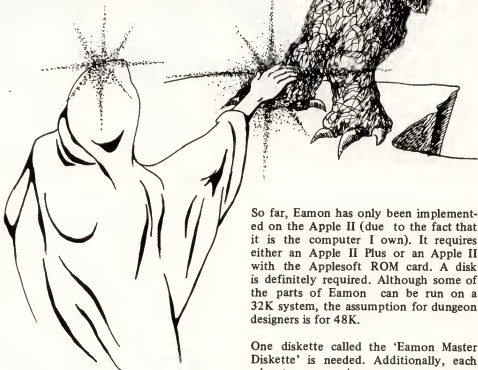
Any player who can hire a wizard to teach him/her can learn these spells. For each spell, a player gets a spell knowledge between 25% and 75%. This is his/her chance of successfully casting the spell the first time. Every time the spell is successfully cast, there is a chance of the spell knowledge going up by 2%, just as with weapon expertise. However, every time a spell is attempted (successfully cast or not!), the chance of casting that spell is cut in half for the rest of the day, down to a minimum of 5% chance of casting. There is also a 1% chance of overloading the brain and forgetting the spell completely every attempt.

For example, assume that Hedric has, through a lucky starting knowledge and some experience, brought his knowledge of a spell to 80%. The first time he

attempts to use the spell during the adventure, his chance of casting it successfully is 80%. Assuming he gets it to work, he has a 20% chance of the spell knowledge going up by 2%. If the spell knowledge does not rise, the next time he tries to use the spell, he only has a 40% chance of success. Unfortunately, he blows it this time and nothing happens. On his third try, his chance of casting the spell is only 20%. If it is successful, his chance of his spell knowledge going up is 80%. Obviously, this means that spell knowledges can go above 100%. Assuming that Hedric's spell ability never rises, the sixth attempt would normally only have a 2% chance of success; however, until the end of the adventure, Hedric's chance of success will remain at 5%.

The first spell that can be learned is the Blast spell. This sends a magic flaming arrow at your target. If the spell is successfully cast, the arrow will always hit the target and do 1D6 points of damage, ignoring armor. The normal cost of the spell is 3000 gold pieces.

The second spell (the ordering is just for convenience; spells may be learned in any order) is Heal. This can only be used on the person casting it, and will



restore 1D10 points of damage. Of course, you cannot cure more points of damage than you have taken. The normal cost is 1000 gold pieces.

The third spell is Speed. This spell will double your Agility for 10 to 35 turns (a turn being the time to do any action). Its major effect is to increase your chance

to hit (remember, Agility affects hit probabilities!) It is not always noticeable when a Speed spell wears off. If the Speed spell is successfully cast again while the first spell holds, the duration of the spell is increased by 10 to 35 turns—your Agility is NOT quadrupled. The normal cost of this spell is 5000 gold pieces.

The last spell is the Power spell, and it is both the most dangerous and yet sometimes the most powerful spell available. There is no set effect of the spell; it is simply a call for the Gods to do something. What they do will vary with time, location, and anything else the dungeon designer wanted to consider. Due to the extreme variability of this spell, it has a normal cost of only 100 gold pieces.

So far, Eamon has only been implemented on the Apple II (due to the fact that it is the computer I own). It requires either an Apple II Plus or an Apple II with the Applesoft ROM card. A disk is definitely required. Although some of the parts of Eamon can be run on a 32K system, the assumption for dungeon designers is for 48K.

One diskette called the 'Eamon Master Diskette' is needed. Additionally, each adventure scenario goes on a separate diskette. At the moment I know of five additional adventure diskettes. Finally, there is a diskette called the 'Dungeon Designer Diskette' that lets people create their own scenarios almost easily. These diskettes are being legitimately freely traded in many places, including the program libraries of computer clubs such as the Apple Pi club of Denver, Colorado.

If you can get copies from such places, it would be preferred. If you cannot find Eamon locally, send an order to Donald Brown, c/o The Computer Emporium, 3711 Douglas Avenue, Des Moines, IA 50310 and include \$5.00 for diskette and postage.

Although many programs are included on the Eamon Master Diskette, the most important is called Main Hall. This program (whose listing is to be found with this article) is where the player can equip himself, find out his/her current weapon expertise and other important information, and head out on individual adventures.

(A word on the DOS used on the Apple for those of you who want to make sense of this program for adapting Eamon to your own uses. Since Applesoft does not have any disk commands built into it, a special kind of software kludge was developed. The DOS is watching all output from the program. If a line begins with a Control-D, the line is assumed to be a disk command and is not printed on the video screen. Two commands that need explaining are READ and WRITE. When a READ command is executed, all of the inputs to the program will come from the file specified, until another disk command is executed. This command can be simply the null command (printing just a Control-D). A WRITE command sends all output of the program into the text file specified instead of to the video screen. In my programs I use a string variable called DK\$ to hold a Control-D, so that any print statements that begin with DK\$ are flagged as disk commands.)

When the MAIN HALL program is entered, there are two important data files expected on the Master Diskette. The first is called THE ADVENTURER. It is a sequential file. It simply has the name of the current player and the record number of the CHARACTERS file where his/her data are stored.

The second important file is called CHARACTERS. It is a random-access file with a length of 150 bytes per record. Record zero holds the total number of records that have been used. In each subsequent record, the first item in the record is a string. If it is the null string (" "), it is a signal to the program that this record is empty and can be used by the NEW CHARACTERS program to place a new player into.

If the first string in a record is not the null string, it is the name of the player. After that are stored the numbers that describe the player—Hardness, Agility, Charisma, the four spell abilities (Blast, Heal, Speed and Power, in that order), and the five weapon abilities (axe, bow,

club, spear, and sword, in that order). The next items are the player's armor expertise, a one-character string that give the player's sex ('M' or 'F'), the number of gold pieces that the character is carrying, the number of gold pieces that the character has deposited with the bank, and a number describing the armor the player is wearing. If the number is odd, the player is carrying a shield. Dividing the number by two and ignoring the remainder gives the armor worn: 0 = none, 1 = leather, 2 = chain, 3 = plate. Finally, the player's four weapons are listed, giving for each the name, weapon type (a number from one to five, signifying which weapon class it belongs to), weapon complexity, the 'x' part of the xDY damage of the weapon, and the 'y' part. If the player is not carrying a weapon, the name of 'NONE' is used and the numbers given with that weapon are irrelevant.

When a player buys or sells something, the reaction of the seller or buyer is determined by coming up with a 'secondary Charisma,' which is gotten by adding or subtracting a random number from zero to five to the player's Charisma. This number is divided by 10, and normal prices are divided by this factor while buying and multiplied by it while selling.

When the player goes on an adventure, he/she is first removed from the CHARACTERS file, so that if he/she doesn't return he/she will be presumed killed. The player is then instructed to switch diskettes. On the adventure diskette a file called EAMON.NAME is opened, and the adventure name is read in (this is why there is only one adventure per disk). The computer then creates a file called FRESH MEAT into which he first writes in the record that the player used to occupy in the CHARACTERS file, then all of the information that was in that record.

After the adventure is finished, the program must have the player switch diskettes back. If the player died, all that must be done is to run 'THE WONDERFUL WORLD OF EAMON,' which will start up a new character. However, if the player survived, the program must re-create the file THE ADVENTURER, restore the player's record in CHARACTERS, with the altered information (hopefully higher weapon expertise and more gold), and run the MAIN HALL program.

To get a feel for EAMON, observe the terrible struggles of Jason of the Boar Ring—

Jason was a starting character with a Hardness of 12, Agility of 21, and Charisma of 14. When he got to the Main Hall, he first decided to buy a weapon.

AS YOU ENTER THE WEAPON SHOP, MARCOS CAVELLIT (THE OWNER) COMES FROM OUT OF THE BACK ROOM AND SAYS, 'WELL, AS I LIVE AND BREATHE, IF IT ISN'T MY OLD PAL JASON! SO, YOU WANT TO BUY A WEAPON, SELL A WEAPON, OR GET SOME BETTER ARMOUR?'

(HIT THE KEY, B OR S OR A) B

MARCOS SMILES AT YOU AND SAYS, 'GOOD. I GOTTA THE BEST. YOU WANT AXE, BOW, MACE, SPEAR, OR SWORD?'

(HIT THE KEYS) A R M B OR S W) SWORD

MARCOS SAYS, 'WELL, I JUST HAPPEN TO HAVE THREE SWORDS IN, OF VARYING QUALITY. I'VE GOT A VERY GOOD ONE FOR 67 GP, A FAIR ONE FOR 33 GP, AND A KINDA SHADY ONE FOR 17 WHICH DO YOU WANT?'

(HIT THE KEYS, G F OR P) G
MARCOS HANDS YOU YOUR WEAPON AND TAKES THE PRICE FROM YOU.

HE NOW ASKS YOU, 'NOW HOW ABOUT SOME ARMOUR?'

(HIT Y OR N) Y
MARCOS TAKES YOU TO THE ARMOUR SECTION OF HIS SHOP AND SHOWS YOU SUITS OF LEATHER ARMOUR, CHAIN ARMOUR, AND PLATE.

HE SAYS, 'I CAN PUT YOU IN ANY OF THESE VERY CHEAPLY. I NEED 67 GOLD PIECES FOR THE LEATHER, 167 FOR THE CHAIN, AND 333 FOR THE PLATE.

WELL, WHAT WILL IT BE?'

(HIT N FOR NOTHING OR L C OR P) N

MARCOS SMILES AND SAYS, 'NOW HOW ABOUT A SHIELD? I CAN LET YOU HAVE ONE FOR ONLY 33 GOLD PIECES!'

(HIT Y OR N) Y

MARCOS TAKES YOUR GOLD AND GIVES YOU A SHIELD.

He also bought a Power spell for 53 gold pieces. Thus, before he went on his first adventure, his character could be described in the following manner:

YOU ARE THE MIGHTY JASON

YOUR ATTRIBUTES ARE:
HD=12 AG=21 CM=14

YOU KNOW THE FOLLOWING SPELLS--
POWER

YOUR WEAPON ABILITIES ARE--
AXE BOW CLUB SPEAR SWORD
5X 10X 20X 10X 0X

ARMOUR: SKIN AND SHIELD AE=0X

COLD IN HAND=47, BANK=0

WEAPONS CMPLX DAM
SWORD 1 D B
(HIT ANY KEY TO CONTINUE)

He went to adventure in the local testing ground called 'The Beginners' Cave,' which is where almost all adventurers first go. He was in a large chamber when things started happening—

YOUR COMMAND?E
YOU ARE IN A SMALL SIZE CHAMBER TO THE EAST OF THE LARGE CHAMBER. IT IS VERY COLD HERE, AND THE ONLY LIGHT COMES IN DIMLY FROM THE LARGE CHAMBER.
THERE IS A GRIZZLED OLD HERMIT WHO 'SMELLS AS IF HE HADN'T TAKEN A BATH IN FORTY YEARS' IN THE ROOM.
THERE IS A BOTTLE HERE WITH A STRANGE POTION INSIDE!

YOUR COMMAND?SILE
NOTHING HAPPENS
YOU ARE STANDING IN
THE EAST-SIDE CHAMBER
HERMIT IS HERE
YOU SEE BOTTLE

(Citizens of Eamon are rather simple-minded; if they like you they will follow you around and attack your enemies. Otherwise, they will attack you!)

. YOUR COMMAND? GET BOTTLE
GOT IT
YOU ARE STANDING IN
THE EAST SIDE CHAMBER
HERMIT IS HERE
YOUR COMMAND? W
YOU ARE STANDING IN
A LARGE CHAMBER
HERMIT IS HERE

YOUR COMMAND? W
YOU ARE IN A SMALL SIDE CHAMBER TO THE WEST OF THE LARGE CHAMBER. IT IS VERY COLD HERE, AND THE ONLY LIGHT COMES IN ONLY FROM THE LARGE CHAMBER.
THERE IS A HUGE BLACK RAT WITH SHARP WHITE TEETH HERE!
THERE IS A HUGE BROWN RAT WITH SHARP WHITE TEETH HERE!
THERE IS A HUGE TAN RAT WITH SHARP WHITE TEETH HERE!
HERMIT IS HERE
THERE IS A PILE OF DIAMONDS HERE!

YOUR COMMAND? ATTACK RAT
JASON ATTACKS BLACK RAT
--A CRITICAL HIT!
TAN RAT IS DEAD!
BLACK RAT ATTACKS JASON
--A MISS
BROWN RAT FLEES OUT AN EXIT.
HERMIT ATTACKS BLACK RAT
--A MISS
YOU ARE STANDING IN
THE WEST SIDE CHAMBER
BLACK RAT IS HERE
HERMIT IS HERE
YOU SEE DIAMONDS
YOU SEE DEAD TAN RAT

YOUR COMMAND? ATTACK RAT
JASON ATTACKS BLACK RAT
--A MISS
BLACK RAT FLEES OUT AN EXIT.
YOU ARE STANDING IN
THE WEST SIDE CHAMBER
HERMIT IS HERE
YOU SEE DIAMONDS
YOU SEE DEAD TAN RAT

(Rats aren't too courageous, are they?)

YOUR COMMAND? GET ALL
DIAMONDS TAKEN
DEAD TAN RAT TAKEN
YOU ARE STANDING IN
THE WEST SIDE CHAMBER
HERMIT IS HERE

He and his new friend the Hermit had no trouble mopping up on the two rats that had escaped them. He also made another friend, a fighter named Heinrich. All was going very well for Jason before he entered this small room--

YOUR COMMAND? E
YOU ARE IN A SMALL, STARK CELL WITH A DOOR ON THE WEST SIDE
HERMIT IS HERE
HEINRICH IS HERE
THERE IS A LARGE CHEST IN THE CENTER OF THE ROOM

YOUR COMMAND? OPEN CHEST
AS YOU APPROACH THE CHEST, IT SUDDENLY BECOMES ALIVE, AND TWO TENTACLES COME OUT OF THE SIDE, HOLDING YOU FAST. ANOTHER TENTACLE RISES FROM THE TOP AND ATTEMPTS TO HURT YOU
YOU ARE STANDING IN
AN EAST CELL
HERMIT IS HERE
HEINRICH IS HERE
A CHEST-SHAPED MONSTER IS IN THE ROOM, HOLDING YOU WITH TWO TENTACLES AND ATTEMPTING TO DISMEMBER YOU WITH A THIRD

YOUR COMMAND? ATTACK CHEST
ATTACK HOW?
YOU ARE STANDING IN
AN EAST CELL
HERMIT IS HERE
HEINRICH IS HERE
MIMIC IS HERE

YOUR COMMAND? ATTACK MIMIC
JASON ATTACKS MIMIC
--A HIT!
MIMIC IS VERY BADLY INJURED
HERMIT ATTACKS MIMIC
--A MISS
HEINRICH ATTACKS MIMIC
--A MISS
MIMIC ATTACKS HERMIT
--A MISS
YOU ARE STANDING IN
AN EAST CELL
HERMIT IS HERE
HEINRICH IS HERE
MIMIC IS HERE

YOUR COMMAND? ATTACK MIMIC
JASON ATTACKS MIMIC
--A MISS
HERMIT ATTACKS MIMIC
--A FUMBLE!
WEAPON DROPPED!
HEINRICH ATTACKS MIMIC
--A MISS
MIMIC ATTACKS HERMIT
--A MISS
YOU ARE STANDING IN
AN EAST CELL
HERMIT IS HERE
HEINRICH IS HERE
MIMIC IS HERE
YOU SEE RING
YOU SEE DEAD MIMIC

YOUR COMMAND? GET ALL
RING TAKEN
DEAD MIMIC IS TOO HEAVY
YOU ARE STANDING IN
AN EAST CELL
HERMIT IS HERE
HEINRICH IS HERE
YOU SEE DEAD MIMIC

Eventually, though, Jason made it out of the adventure with his treasures intact. He sold them and had himself analyzed again. This time, his attributes and abilities were as follows:

(HIT THE KEY FOR YOUR CHOICE, 1-635)
YOU ARE THE MIGHTY JASON
YOUR ATTRIBUTES ARE
HE=12 AG=21 CH=14
YOU KNOW THE FOLLOWING SPELLS--
POWER
YOUR WEAPON ABILITIES ARE--
RIVE BOW CLUB SPEAR SHORDE
% -10% 20% 10% 6%
ARMOUR SKIN AND SHIELD RE=6%
GOLD IN HAND=577, GRW=0
WEAPONS CMLX DRM
SHORDE 10% 1 D 6
RIVE 10% 1 D 6

Notice that through experience both sword ability and armor expertise went up by 6%, or increased three times. Until Jason buys heavier armor, his armor expertise cannot increase any more. He also has 657 gold pieces, which should permit him to buy a new spell.

Although Eamon has a reasonably complete background set, there is still room for infinite expansion. Jason may have been able to handle small problems like the rats and survive little traps like the chest-mimic, but surely in the depths of your imagination you have nastier things in mind! Furthermore, so far all of the adventures written for Eamon

have resembled the Woods and Crowther's ADVENTURE, but Eamon is more flexible than that. Consider a game where you are leading an army into battle, with morale affected by your Charisma! The possibilities are endless. There is a full universe out there, and it is up to you to pull it into your computer!



Donald Brown became a personal computing fanatic almost three years ago when his father brought home one of the first Apple II's. He also permitted the infamous William Fesselmeyer to lure him into fantasy games over two years ago. He now spends his time in Des Moines writing computer programs, playing games, and incidentally working towards a Bachelor's Degree at Drake University. He can best be contacted through the Computer Emporium of Des Moines.

```

00 REM THE WONDERFUL WORLD OF
01 REM EMMON
02 REM
03 REM MAIN HALL <BUY SPELLS
04 REM AND EQUIPMENT AND
05 GO ON ADVENTURES>
06 REM
07 REM BY DONALD BROWN
08 REM
09 REM
10 REM CHR$(4)
11 ONOPR GOTO 7000
12 PRINT DEF: "OPEN THE ADVENTURE
13 P.56, DEF: "PRINT DEF: "READ TH
14 DEF: "READ DEF: "WRITE: INPUT
15 DEF: PRINT DEF: "CLOSE"
16 REM
17 ONOPR GOTO 100
18 PRINT DEF: "OPEN CHARACTERS.L1
19 50": PRINT DEF: "READ CHARACTER
20 ERS.P": REC: INPUT NAME$: INPUT
21 NO.AC.CH: FOR X = 1 TO 4: INPUT
22 NAME$(X): FOR X = 1 TO 4: INPUT
23 NAME$(X): NEXT X: INPUT
24 NAME$(X): NEXT X: INPUT
25 NAME$(X): NEXT X: INPUT
26 REM
27 INPUT SEX:
28 INPUT GOLD.BANK: INPUT AC: FOR
29 X = 1 TO 4: INPUT NAME$(X):
30 NAME$(X), ADDRESS$(X), ADDRESS$(X)
31 NAME$(X), NEXT X: PRINT
32 DEF: "CLOSE"
33 DEF: FN P(X) = INT (X / RTIO
34 + 5)
35 PRINT : PRINT "AS YOU WALKER
36 ABOUT THE HALL, YOU": PRINT
37 "REALIZE YOU CAN DO ONE OF 5
38 THINGS:
39 1. GO ON AN ADVENTURE" PRINT
40 2. VISIT THE HERFON SHOP"
41 3. BUY WEAPONS AND/OR ARM
42 OUR"
43 PRINT "3. HIRE A FINDER. T
44 O TEACH YOU SOME A FINDER. T
45 LLS": PRINT POSIT OF PRINT
46 "BRIEFER: INPUT DEF: "PRINT
47 WITHDRAW SOME GOLD":
48 PRINT "5. EXAMINE YOUR A
49 BILITIES": PRINT "6. TEN
50 POSSIBLY LEAVE THE UNIVERSE"
51 PRINT
52 PRINT "WHAT THE KEY
53 YOUR CHANCE:
54 GET A CHD = VAL (DEF: IF C
55 NO < 1.0P CHD : 5 THEN 130
56 PRINT CHD: PRINT "OH GOOD! GOTO
57 10000.30000.40000.50000.600
58 0
59 PRINT NAME$: PRINT NO: PRINT
60 NAME$(X): FOR X = 1 TO 4:
61 PRINT NAME$(X): NEXT X:
62 4. TO 5: PRINT NAME$(X): NEXT
63 X: PRINT DEF: "CLOSE"
64 PRINT GOLD: PRINT BANK: PRINT AN
65 AL FOR W = 1 TO 4: PRINT AN
66 NAME$(X): PRINT NAME$(X): PRINT
67 NAME$(X): PRINT NAME$(X): PRINT
68 NAME$(X): PRINT NAME$(X): PRINT
69 NAME$(X): PRINT NAME$(X): PRINT
70 NAME$(X): PRINT NAME$(X): PRINT
71 NAME$(X): PRINT NAME$(X): PRINT
72 NAME$(X): PRINT NAME$(X): PRINT
73 NAME$(X): PRINT NAME$(X): PRINT
74 NAME$(X): PRINT NAME$(X): PRINT
75 NAME$(X): PRINT NAME$(X): PRINT
76 NAME$(X): PRINT NAME$(X): PRINT
77 NAME$(X): PRINT NAME$(X): PRINT
78 NAME$(X): PRINT NAME$(X): PRINT
79 NAME$(X): PRINT NAME$(X): PRINT
80 NAME$(X): PRINT NAME$(X): PRINT
81 NAME$(X): PRINT NAME$(X): PRINT
82 NAME$(X): PRINT NAME$(X): PRINT
83 NAME$(X): PRINT NAME$(X): PRINT
84 NAME$(X): PRINT NAME$(X): PRINT
85 NAME$(X): PRINT NAME$(X): PRINT
86 NAME$(X): PRINT NAME$(X): PRINT
87 NAME$(X): PRINT NAME$(X): PRINT
88 NAME$(X): PRINT NAME$(X): PRINT
89 NAME$(X): PRINT NAME$(X): PRINT
90 NAME$(X): PRINT NAME$(X): PRINT
91 NAME$(X): PRINT NAME$(X): PRINT
92 NAME$(X): PRINT NAME$(X): PRINT
93 NAME$(X): PRINT NAME$(X): PRINT
94 NAME$(X): PRINT NAME$(X): PRINT
95 NAME$(X): PRINT NAME$(X): PRINT
96 NAME$(X): PRINT NAME$(X): PRINT
97 NAME$(X): PRINT NAME$(X): PRINT
98 NAME$(X): PRINT NAME$(X): PRINT
99 NAME$(X): PRINT NAME$(X): PRINT
100 NAME$(X): PRINT NAME$(X): PRINT

```

[illegible]

```

22500 PRINT R#; PRINT "P = " THEN PRINT
22510 IF R# < 50 / RTIO + 50; PRINT "HRC
22520 OS SAYS, "THEN I CAN ONLY GI
22530 VE YOU," THEN P, " GOLD PIEC
22540 ES FOR IT, TAKE IT OR," PRINT
22550 "LEAVE IT," THEN PRINT
22560 GET R#; IF R# < 2210 THEN PRINT
22570 "NEXT M"
22580 IF R# < "T" THEN 22590
22590 PCOS SAYS, "OKAY, HART-VE VO
22600 GOS SAYS, "GOLD PIECE," THEN
22610 "LEAVE IT," THEN PRINT
22620 IF R# < 2210 THEN PRINT
22630 "NEXT M"
22640 IF R# < "T" THEN 22650
22650 PCOS SAYS, "OKAY, HART-VE VO
22660 GOS SAYS, "GOLD PIECE," THEN
22670 "LEAVE IT," THEN PRINT
22680 IF R# < 2210 THEN PRINT
22690 "NEXT M"
22700 IF R# < "T" THEN 22710
22710 PCOS SAYS, "OKAY, HART-VE VO
22720 GOS SAYS, "GOLD PIECE," THEN
22730 "LEAVE IT," THEN PRINT
22740 IF R# < 2210 THEN PRINT
22750 "NEXT M"
22760 IF R# < "T" THEN 22770
22770 PCOS SAYS, "OKAY, HART-VE VO
22780 GOS SAYS, "GOLD PIECE," THEN
22790 "LEAVE IT," THEN PRINT
22800 IF R# < 2210 THEN PRINT
22810 "NEXT M"
22820 IF R# < "T" THEN 22830
22830 PCOS SAYS, "OKAY, HART-VE VO
22840 GOS SAYS, "GOLD PIECE," THEN
22850 "LEAVE IT," THEN PRINT
22860 IF R# < 2210 THEN PRINT
22870 "NEXT M"
22880 IF R# < "T" THEN 22890
22890 PCOS SAYS, "OKAY, HART-VE VO
22900 GOS SAYS, "GOLD PIECE," THEN
22910 "LEAVE IT," THEN PRINT
22920 IF R# < 2210 THEN PRINT
22930 "NEXT M"
22940 IF R# < "T" THEN 22950
22950 PCOS SAYS, "OKAY, HART-VE VO
22960 GOS SAYS, "GOLD PIECE," THEN
22970 "LEAVE IT," THEN PRINT
22980 IF R# < 2210 THEN PRINT
22990 "NEXT M"
23000 IF R# < "T" THEN 23010
23010 PCOS SAYS, "OKAY, HART-VE VO
23020 GOS SAYS, "GOLD PIECE," THEN
23030 "LEAVE IT," THEN PRINT
23040 IF R# < 2210 THEN PRINT
23050 "NEXT M"
23060 IF R# < "T" THEN 23070
23070 PCOS SAYS, "OKAY, HART-VE VO
23080 GOS SAYS, "GOLD PIECE," THEN
23090 "LEAVE IT," THEN PRINT
23100 IF R# < 2210 THEN PRINT
23110 "NEXT M"
23120 IF R# < "T" THEN 23130
23130 PCOS SAYS, "OKAY, HART-VE VO
23140 GOS SAYS, "GOLD PIECE," THEN
23150 "LEAVE IT," THEN PRINT
23160 IF R# < 2210 THEN PRINT
23170 "NEXT M"
23180 IF R# < "T" THEN 23190
23190 PCOS SAYS, "OKAY, HART-VE VO
23200 GOS SAYS, "GOLD PIECE," THEN
23210 "LEAVE IT," THEN PRINT
23220 IF R# < 2210 THEN PRINT
23230 "NEXT M"
23240 IF R# < "T" THEN 23250
23250 PCOS SAYS, "OKAY, HART-VE VO
23260 GOS SAYS, "GOLD PIECE," THEN
23270 "LEAVE IT," THEN PRINT
23280 IF R# < 2210 THEN PRINT
23290 "NEXT M"
23300 IF R# < "T" THEN 23310
23310 PCOS SAYS, "OKAY, HART-VE VO
23320 GOS SAYS, "GOLD PIECE," THEN
23330 "LEAVE IT," THEN PRINT
23340 IF R# < 2210 THEN PRINT
23350 "NEXT M"
23360 IF R# < "T" THEN 23370
23370 PCOS SAYS, "OKAY, HART-VE VO
23380 GOS SAYS, "GOLD PIECE," THEN
23390 "LEAVE IT," THEN PRINT
23400 IF R# < 2210 THEN PRINT
23410 "NEXT M"
23420 IF R# < "T" THEN 23430
23430 PCOS SAYS, "OKAY, HART-VE VO
23440 GOS SAYS, "GOLD PIECE," THEN
23450 "LEAVE IT," THEN PRINT
23460 IF R# < 2210 THEN PRINT
23470 "NEXT M"
23480 IF R# < "T" THEN 23490
23490 PCOS SAYS, "OKAY, HART-VE VO
23500 GOS SAYS, "GOLD PIECE," THEN
23510 "LEAVE IT," THEN PRINT
23520 IF R# < 2210 THEN PRINT
23530 "NEXT M"
23540 IF R# < "T" THEN 23550
23550 PCOS SAYS, "OKAY, HART-VE VO
23560 GOS SAYS, "GOLD PIECE," THEN
23570 "LEAVE IT," THEN PRINT
23580 IF R# < 2210 THEN PRINT
23590 "NEXT M"
23600 IF R# < "T" THEN 23610
23610 PCOS SAYS, "OKAY, HART-VE VO
23620 GOS SAYS, "GOLD PIECE," THEN
23630 "LEAVE IT," THEN PRINT
23640 IF R# < 2210 THEN PRINT
23650 "NEXT M"
23660 IF R# < "T" THEN 23670
23670 PCOS SAYS, "OKAY, HART-VE VO
23680 GOS SAYS, "GOLD PIECE," THEN
23690 "LEAVE IT," THEN PRINT
23700 IF R# < 2210 THEN PRINT
23710 "NEXT M"
23720 IF R# < "T" THEN 23730
23730 PCOS SAYS, "OKAY, HART-VE VO
23740 GOS SAYS, "GOLD PIECE," THEN
23750 "LEAVE IT," THEN PRINT
23760 IF R# < 2210 THEN PRINT
23770 "NEXT M"
23780 IF R# < "T" THEN 23790
23790 PCOS SAYS, "OKAY, HART-VE VO
23800 GOS SAYS, "GOLD PIECE," THEN
23810 "LEAVE IT," THEN PRINT
23820 IF R# < 2210 THEN PRINT
23830 "NEXT M"
23840 IF R# < "T" THEN 23850
23850 PCOS SAYS, "OKAY, HART-VE VO
23860 GOS SAYS, "GOLD PIECE," THEN
23870 "LEAVE IT," THEN PRINT
23880 IF R# < 2210 THEN PRINT
23890 "NEXT M"
23900 IF R# < "T" THEN 23910
23910 PCOS SAYS, "OKAY, HART-VE VO
23920 GOS SAYS, "GOLD PIECE," THEN
23930 "LEAVE IT," THEN PRINT
23940 IF R# < 2210 THEN PRINT
23950 "NEXT M"
23960 IF R# < "T" THEN 23970
23970 PCOS SAYS, "OKAY, HART-VE VO
23980 GOS SAYS, "GOLD PIECE," THEN
23990 "LEAVE IT," THEN PRINT
24000 IF R# < 2210 THEN PRINT
24010 "NEXT M"
24020 IF R# < "T" THEN 24030
24030 PCOS SAYS, "OKAY, HART-VE VO
24040 GOS SAYS, "GOLD PIECE," THEN
24050 "LEAVE IT," THEN PRINT
24060 IF R# < 2210 THEN PRINT
24070 "NEXT M"
24080 IF R# < "T" THEN 24090
24090 PCOS SAYS, "OKAY, HART-VE VO
24100 GOS SAYS, "GOLD PIECE," THEN
24110 "LEAVE IT," THEN PRINT
24120 IF R# < 2210 THEN PRINT
24130 "NEXT M"
24140 IF R# < "T" THEN 24150
24150 PCOS SAYS, "OKAY, HART-VE VO
24160 GOS SAYS, "GOLD PIECE," THEN
24170 "LEAVE IT," THEN PRINT
24180 IF R# < 2210 THEN PRINT
24190 "NEXT M"
24200 IF R# < "T" THEN 24210
24210 PCOS SAYS, "OKAY, HART-VE VO
24220 GOS SAYS, "GOLD PIECE," THEN
24230 "LEAVE IT," THEN PRINT
24240 IF R# < 2210 THEN PRINT
24250 "NEXT M"
24260 IF R# < "T" THEN 24270
24270 PCOS SAYS, "OKAY, HART-VE VO
24280 GOS SAYS, "GOLD PIECE," THEN
24290 "LEAVE IT," THEN PRINT
24300 IF R# < 2210 THEN PRINT
24310 "NEXT M"
24320 IF R# < "T" THEN 24330
24330 PCOS SAYS, "OKAY, HART-VE VO
24340 GOS SAYS, "GOLD PIECE," THEN
24350 "LEAVE IT," THEN PRINT
24360 IF R# < 2210 THEN PRINT
24370 "NEXT M"
24380 IF R# < "T" THEN 24390
24390 PCOS SAYS, "OKAY, HART-VE VO
24400 GOS SAYS, "GOLD PIECE," THEN
24410 "LEAVE IT," THEN PRINT
24420 IF R# < 2210 THEN PRINT
24430 "NEXT M"
24440 IF R# < "T" THEN 24450
24450 PCOS SAYS, "OKAY, HART-VE VO
24460 GOS SAYS, "GOLD PIECE," THEN
24470 "LEAVE IT," THEN PRINT
24480 IF R# < 2210 THEN PRINT
24490 "NEXT M"
24500 IF R# < "T" THEN 24510
24510 PCOS SAYS, "OKAY, HART-VE VO
24520 GOS SAYS, "GOLD PIECE," THEN
24530 "LEAVE IT," THEN PRINT
24540 IF R# < 2210 THEN PRINT
24550 "NEXT M"
24560 IF R# < "T" THEN 24570
24570 PCOS SAYS, "OKAY, HART-VE VO
24580 GOS SAYS, "GOLD PIECE," THEN
24590 "LEAVE IT," THEN PRINT
24600 IF R# < 2210 THEN PRINT
24610 "NEXT M"
24620 IF R# < "T" THEN 24630
24630 PCOS SAYS, "OKAY, HART-VE VO
24640 GOS SAYS, "GOLD PIECE," THEN
24650 "LEAVE IT," THEN PRINT
24660 IF R# < 2210 THEN PRINT
24670 "NEXT M"
24680 IF R# < "T" THEN 24690
24690 PCOS SAYS, "OKAY, HART-VE VO
24700 GOS SAYS, "GOLD PIECE," THEN
24710 "LEAVE IT," THEN PRINT
24720 IF R# < 2210 THEN PRINT
24730 "NEXT M"
24740 IF R# < "T" THEN 24750
24750 PCOS SAYS, "OKAY, HART-VE VO
24760 GOS SAYS, "GOLD PIECE," THEN
24770 "LEAVE IT," THEN PRINT
24780 IF R# < 2210 THEN PRINT
24790 "NEXT M"
24800 IF R# < "T" THEN 24810
24810 PCOS SAYS, "OKAY, HART-VE VO
24820 GOS SAYS, "GOLD PIECE," THEN
24830 "LEAVE IT," THEN PRINT
24840 IF R# < 2210 THEN PRINT
24850 "NEXT M"
24860 IF R# < "T" THEN 24870
24870 PCOS SAYS, "OKAY, HART-VE VO
24880 GOS SAYS, "GOLD PIECE," THEN
24890 "LEAVE IT," THEN PRINT
24900 IF R
```



```

3840 GET A# FOR SP = 1 TO 4: IF
A# < MID$( "BMSPP", SP, 1) THEN
NEXT: GOTO 3840
3850 ON SP GOTO 3860, 3870, 3880, 3
3860 PRINT "BLST": P = FN P(388
0: GOTO 3400
3870 PRINT "HEAL": P = FN P(4000
) GOTO 3400
3880 PRINT "SPEED": P = FN P(5000
0: GOTO 3400
3890 PRINT "POWER": P = FN P(1000
)
3400 PRINT: IF P > GOLD THEN PRINT
"WHEN HOKAS SEES THAT YOU DO
N'T HAVE", PRINT "ENOUGH TO
PAY HIM", HE STALKS TO THE FROGS
TERS WHO: PRINT "SHOULD BE
TURNED INTO FROGS": PRINT:
GOTO 100
3410 IF SAK(SP) THEN PRINT "HOK
AS SAYS, 'DON'T RISK IT TO
US', PRINT "GOD KNOWS", BUT
HAVEN'T YOU FORGOTTEN: PRINT
"SOMETHING? I ALREADY TROU
GH YOU THAT: PRINT "SPELL!"
PRINT: PRINT "SHAKING HIS
HEAD, SAKU HE RETURNS TO:
PRINT "BRR": PRINT: GOTO
100
3420 GOLD = GOLD - P: SAK(SP) = INT
(50 + RND(1) * 26): PRINT
"HOKAS TEACHES YOU YOUR SPELL
CANNOT BE USED: PRINT "THIS SPELL A
NO HARMFUL TRICKS TO HIS GOLD":
PRINT "THE BRR AS YOU WAL
K AWAY YOU HEAR"
3430 PRINT "HIM ORDER A DOUBLE D
RAGON BOMB": PRINT: GOTO
100
3440
3450 PRINT: PRINT "YOU HAVE NO
TROUBLE SPOTTING SHYLOCK": PRINT
"CFENNEY, THE LOCAL BANKER,
DUE TO HIS": PRINT "LARGE B
ELLY: PRINT "YOU A
ELLY: PRINT "HITTING THE
COMES": PRINT "OVER TO YOU
AND SAYS: 'WELL,"
4010 PRINT NAME#, MY DEAR ", MID$
("GRLBOY", (SEX# = "M") * 4 +
1, 4), "": PRINT "WHAT A PLEA
SURE TO SEE YOU TO YOU WANT
4020 PRINT "TO MAKE A DEPOSIT OR
A WITHDRAWAL?": PRINT: PRINT
"CHIT-D FOR DEPOSIT, 'M' F
OR WITHDRAWAL": PRINT
4030 IF PRINT: GOTO 4040
4040 IF A# < 4 THEN PRINT
4050 PRINT A#: PRINT: PRINT "SH
YLOCK SAYS: 'WELL, YOU HAVE
BROCK', PRINT "GOLD PIECES S
TO YOU WANT TO TAKE BROCK?":
PRINT

```

```

4060 PRINT "ENTER THE NUMBER OF
GOLD PIECES TO", INPUT
WITHDRAW", A# RNV = VAL
A#): IF RNV < 0 OR RNV >
INT (RNV) OR STR# (RNV) <
HE BARKER SCOWLS AND SAYS: "C
OME, COME": PRINT "YOU'RE N
OT MAKING SENSE! TRY AGAIN
": PRINT: GOTO 4060
4070 IF RNV > BNV THEN PRINT:
PRINT "THAT'S MORE THAN
U A TERIBLE GUARTEE: PRINT
"AND SAYS: 'THAT'S MORE THAN
YOU'VE GOT": PRINT "YOU KN
OW I DON'T HAVE LONS TO YOU
THE PRINT "KIND", WITH THAT
HE BARKER SCOWLS AND SAYS: "C
OME, COME": PRINT "YOU'RE N
OT MAKING SENSE! TRY AGAIN
": PRINT: GOTO 100
4080 GOLD = GOLD + RNV: BNV = BAN
KER HANDS YOU YOUR GOLD A
BARKER SAYS: "THAT LEAV
ES YOU IN MY CAFE: HE SHAKE
S YOUR": PRINT "HAND AND WAL
KS AWAY": PRINT: GOTO 100
4090 PRINT "SHYLOCK GETS A WIDE
GRIN ON HIS FACE: PRINT "AN
OTHER DAY, SHYLOCK GETS ON
MUCH CO": PRINT "YOU WANT TO
DEPOSIT?":
4100 PRINT: PRINT "ENTER THE A
MOUNT YOU WANT TO DEPOSIT":
INPUT A# IF RNV < 0 OR RNV >
INT (RNV) OR STR# (RNV) <
A# THEN PRINT: PRINT "T
HE BARKER SCOWLS AT YOU AND
SAYS: 'PRINT "COME, COME, PRINT
"YOU'RE NOT MAKING SENSE! TRY
AGAIN": PRINT: GOTO
4100
4110 IF RNV > GOLD THEN PRINT:
PRINT "THE BARKER WAS VERY
PLEASD WHEN YOU: PRINT "TO
DO HIM THE SUM, BUT WHEN HE
HE BARKER SCOWLS AND SAYS: "C
OME, COME": PRINT "YOU'RE N
OT MAKING SENSE! TRY AGAIN
": PRINT: GOTO 100
4120 GOLD = GOLD - RNV: BNV = BAN
KER HANDS YOU YOUR GOLD A
BARKER SAYS: "THAT LEAV
ES YOU IN MY CAFE: HE SHAKE
S YOUR": PRINT "HAND AND WAL
KS AWAY": PRINT: GOTO 100

```

```

5000 HOME: PRINT "YOU ARE THE "
MIDE "MIGHTY FAIR", 1 + 7
* (SEX# = "F") * 7, NAMES: ARE
": PRINT "YOUR ATTRIBUTES ARE
": PRINT "HD="HD": AG="
AG": CH="CH": PRINT
5010 PRINT "YOU KNOW THE FOLLOWI
NG SPELLS--", SM = 0: FOR S =
1 TO 4: IF SAK(S) THEN PRINT
MID$ ("BLASTHEAL SPEEDPOWER
", S) - 1 * 5 + 4 * S": S
NEXT S: IF NOT SM THEN PRINT
"NO SPELLS,"
5020 PRINT: PRINT "YOUR
WEAPON ABILITIES ARE--": PRINT
"AXE BOON CLUB SPEAR S
WORD SPINT SPEC FOR W:
W < 0 * 5: PRINT "W:
W < 0 * 4 * ABS (WAK(W) < 100)
0 * 4 * ABS (WAK(W) < 100)
JMAK(W) % ": NEXT: PRINT
5040 PRINT: A2 = INT (AC / 2): PRINT
"ARMOUR", MID$ ("SKIN LE
ATHERCHAIN PLATE ", A2 * 7 +
1, 7): IF AC > A2 + 2 THEN PRINT
"AND SHIELD":
5050 PRINT TRK(38): "REH" AE "H
": PRINT "GOLD IN HA
ND="J, GOLD": BNV="BNV":
5060 PRINT: PRINT "WEAPONS", TRK(
19): "CNFLX", DAN": FOR W =
1 TO 4: PRINT "W:
"NONE" THEN 5090 AND WAK(W) =
5070 IF WAK(W) = "NONE" THEN PRINT
: PRINT "NO WEAPONS"
5080 GOTO 5090
5090 PRINT WAK(W), TRK(50) - 40
W < 0 * 4 * ABS (WAK(W) < 100)
10 * 4 * ABS (WAK(W) < 100)
JMAK(W) % ":
JMAK(W) %
5090 NEXT W: VTRB 23: POKE - 15
368, 0: PRINT "CHIT MAY MEV T
CHARACTERS": GET A#: PRINT
: GOTO 100
5100 END
5000 PRINT "AS YOU LEAVE THE HAL
L, THE IRISHMAN: PRINT "COM
E UP TO YOU, SLAPS YOU ON T
HE BACK AND SAYS: 'WELL, YOU'VE
WALKED BACK REAL SOON":
PRINT "YR HEAR?":
5010 PRINT DK#, "OPEN CHARACTERS,
L150, S6-D1": PRINT D# #, HPRT
E CHARACTERS, R: RECI: GOSUB 2
00: PRINT DK#, "CLOSE"
5020 PRINT DK#, "DELETE THE ADVEN
Turer"
5030 END
5040 PEEK (<22) = 0 THEN PRINT
IF NOT THE
WORLD OF ERMON"
5050 END

```

The End

```

2629 M=23 => EG
2630 PRINT "a solid egg is on the floor." : RETURN
2631
2632 M=24 => JEWEL
2633 PRINT "on the floor is a JEWEL." : RETURN
2634
2635 M=25 => URSE
2636 PRINT "a magnificent URSE is here." : RETURN
2637
2638 --- responding to listed words ---
2639
2640 "You could set S=0 here & switch S to one if the message
2641 were understood. You'd have 3995 ON S+1 GOTO 365. 385
2642
2643 IF R#="EXTING TIME" THEN T9 = T9+24:R#>0 GOTO 5315
2644
2645 "GOTO 565
2646
2647 --- details of settings & droppings ---
2648
2649 "Assign M an appropriate value. (M is the # of the
2650 manifestation named by H. Unless M = 0).
2651 IF US OR HS THEN M#0 RETURN
2652 IF M#24 THEN IF L(12)>9 THEN M#11 RETURN ELSE M#14 RETURN
2653 FOR J=1 TO M9 IF L(14)=J THEN M#1 RETURN ELSE NEXT J
2654 FOR M#1 IF M#22 AND M#12 THEN RETURN
2655 FOR J=3 TO 5: IF L(3)=J THEN M#J : NEXT J ELSE NEXT J
2656 RETURN
2657
2658 "GOTO 565
2659
2660 IF L(10)=R AND C<9 THEN 4875 ELSE IF C>9 THEN 4055
2661 PRINT "You're still set at the 'HIS' GOTO 4140
2662
2663 "You can only carry 'J C9' objects (not counts)
2664 IF US OR HS AND C#0 THEN PRINT "that such."
2665 PRINT "If you really want TO GET something else, then you
2666 will have to DROP something first." : GOTO 4140
2667
2668 IF M#10 THEN 4090 ELSE PRINT "Congratulations." : P#P(9)
2669 PRINT "You'll earn points when the 'HIS' is DEPOSITED (or
2670 dropped) in the Bank." : C#C1 : L(10)=1 : RETURN
2671
2672 "If you are now carrying the 'J H8' :
2673 L(10) = 1 : C#C1 : RETURN
2674
2675 IF L(9) = -1 THEN PRINT "G-." : L(N)=R : R=C-1 ELSE 4135
2676 FOR R=0 THEN GOSUB 5175
2677 IF R<3 AND R<9 THEN RETURN
2678 IF L(2)=3 THEN L(2)=0 : L(3)=0 : L(7)=0 : L(10)=3
2679 IF L(4)=9 THEN L(4)=0 : L(3)=0 : L(11)=0 : L(12)=9
2680 IF L(5)=9 THEN L(5)=0 : L(3)=0 : L(11)=0 : L(12)=9
2681 RETURN
2682
2683 "You don't have the 'J H8' :
2684
2685 PRINT "see an inventory of what you're carrying. Sure"
2686 PRINT "INVENTORY." : RETURN
2687
2688 --- details of UP and DOWN moving ---
2689
2690 "This block is not used in the Mellan is Thirsty
2691 version of Enchanted House. For use, see the
2692 July-Aug. 1980 issue of RECREATIONAL COMPUTING.
2693
2694 "Setting H (# for floor that we're on)
2695 IF R#7#8 : FOR H=1 TO 8 : (top floor same with H=1)
2696 IF H#1-2 : R AND R#12 THEN 4325 ELSE NEXT H
2697 "checking for special conditions common to U & D
2698 GOTO 4070
2699 IF H#1 OR H#8 : L#R#C1-1) : L#R#C1-1)
2700 IF L(10)=0 THEN 4405 ELSE 4455
2701 "UPing
2702 IF H#1 THEN 4445
2703 "code such as
2704 "similar to the code found in lines 265- 290.
2705 PRINT "THE CHRIS is too heavy to carry UP." GOTO 385
2706 R = R-7#8 : GOTO 405
2707 PRINT "There's no way to GO UP here." GOTO 305
2708 "GOing
2709 "This part follows the same pattern as indicated in
2710 the UPing part above.
2711
2712 --- drawing map ---
2713
2714 I=0 : PRINT "PRINT "You are in the 'H#4#(R)'. Room now." : J
2715 PRINT TAB(11) "PRINT TAB(11-2) "J" = E"
2716 PRINT TAB(11) "J"
2717
2718 IF I#7#8 : FOR I=1 TO 86
2719 IF I#1 THEN IF H#1 THEN 4505 ELSE GOSUB 4500 THEN RETURN
2720 IF I#11-12 : R AND R#12 THEN 4530 ELSE NEXT I
2721
2722 R = (I-1)-12 : R = I-12 : I#12 : I#12 : I#12
2723
2724 IF K#0 AND J#1 THEN PRINT U$(K+7)
2725 IF K#0 AND J#1 AND J#8#1 THEN PRINT U$(K+8)
2726 IF K#0 AND J#8#1 THEN PRINT U$(K+9)
2727
2728 IF K#0 AND K#9 AND J#1 THEN PRINT U$(K+4)
2729 IF K#0 AND K#9 AND J#1 AND J#8#1 THEN PRINT U$(K+5)
2730 IF K#0 AND K#9 AND J#8#1 THEN PRINT U$(K+6)
2731
2732 IF K#5 AND J#1 THEN PRINT U$(K+1)
2733 IF K#9 AND J#1 AND J#8#1 THEN PRINT U$(K+2)
2734 IF K#9 AND J#8#1 THEN PRINT U$(K+3)
2735
2736 IF J#8 THEN 4525
2737 IF M#0, J#0 THEN PRINT M$(K+K+J) ELSE PRINT M$(K+J)
2738 J+1 : GOTO 4535
2739
2740 IF J#9 THEN RETURN ELSE J=1
2741
2742 IF M#0, J#0 THEN PRINT M$(K+K+J) ELSE PRINT M$(K+J)
2743 IF J#8 THEN PRINT GOTO 4620
2744
2745 IF U$(K+K+J) THEN PRINT U$(K+K+J) ELSE PRINT U$(K+J)
2746 J+1 : GOTO 4600
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

```

```

1250 PRINT "Not sure what the clues mean!" --- "I
1255 PRINT "think & experiment." GOTO 505
1260
1265 IF you like THEN for you to fill in
1270
1275 for defaulting
1280
1285 PRINT "The 'N' is hard to 'N' 'N' " I GOTO 505
1290
----- listing variables -----
1300
1305 Jm, variables = 1, J, R, 19, 11, & a few others
1310 M is provincial to the 4268-4499 block.
1315
1320 U3 = # of verbs understood
1325 U4 = # of nouns understood
1330 U5 = # for verb of command
1335 N = # for noun of command
1340
1345 M9 = # of manifestations
1350 M = # of a manifestation
1355 L10 = location of manifestation M where
1360 L11 = # 5 means object 5 is in room 5.
1365 L14 = -1 means object 4 is being carried by the player.
1370 L15 = # 0 means object 6 is in the inaccessible room
1375 U10 = worth of manifestation M (worth in points)
1380
1385 W1C tells of N-S passages
1390 W2C tells of E-W passages where
1395 values of W1C & W2C have the following meanings:
1400 1 will present (not really a passage at all)
1405 2 passage tried & clear 1 no demands or changes
1410 2.5 demands or changes, clear last time
1415 3.5 demands or changes, blocked last time
1420 negative 3 means untied, finding means positive
1425
1430 N
1435 W S Rules handled by W2(R) (room R) W2(R+1)
1440 W1(R+8)
1445
1450 U1(R), U2(R), & U3(R) have to do with drawing the map.
1455 U3(R) is an upper left corner --- see keypad for sense
1460
1465 W4(R) = 1 if R has been visited & 0 otherwise
1470 W4(R) used in naming rooms on the map
1475
1480 R6 = # of floors in the house
1485 R7 = # of rows of rooms on each floor
1490 R8 = # of columns of rooms on each floor
1495 R9 = # of rooms = R6*R7*R8
1500 R10 = # of initial room = # of Bank room = R9-R8+1
1505 R = # of current room
1510
1515 P5 = # of Players (but just one at a time)
1520 P = # of the current player
1525 P1(P) = name of current player
1530
1535 T = total # of passages made
1540 T1 = limit on T
1545 T2 = # passages this turn of current player
1550 T3 = limit on T1
1555
1560 Q9 = # questions asked in soliciting player's choices
1565 Q10 holds questions
1570
1575 F5 = # of faces for the Banker
1580 B1 holds faces of Banker
1585
1590 C = # items carried including map but excluding watch
1595 C1 = limit on C
1600
1605 M1(N) = N means the N th noun describes manifestation M
1610
1615 ECG, YG, NG are used only for Heath graphics
1620
----- listing of verbs, nouns, and manifestations -----
1625
1630 Verb I noun I manifestation I N1(C)
1635
1640 TAKE H 1 catalog 20
1645 GET S 2 coupon 13
1650 GRAB E 1 empty bowl 12
1655 PICK W 4 bowl with cold milk 12
1660
1665 LPOP NORTH 5 bowl with warm milk 12
1670 GIVE SOUTH 6 carrot food 9
1675 OFFER EAST 7 machine with bowls 0
1680 DEPOSIT WEST 8 machine without bowls 0
1685
1690 KICK DOOR 9 banker 23
1695 CONSULT CHAIR 10 key 16
1698 UNLOCK RABBIT 11 cat telling of wanting milk 24
1700 UNLOCK BOWL 12 cat with warm milk 24
1705
1710 FILL COUPON 13 cat with cold milk 24
1715 LIST WRAPPER 14 naming cat 24
1720 WORTH MAP 15 usual rabbit 24
1725 SIT KEY 16 rabbit giving egg 11
1730
1735 DRINK EGG 17 rabbit eating carrot door 11
1740 HELP OPEN 18 initial rabbit missing 11
1745 GO JEWEL 19 magic map 15
1750 FELLOW CATALOG 20 wrapper 14
1755
1760 USE 21 voice giving reference 0
1765 MILK 22 coin 17
1770 BANKER 23 egg 18
1775 CAT 24 jewel 19
1780
1785 WORDS 25 use 21
1790 INVENTORY 26
1795 SCORE 27
1800 WATCH 28
1805
1810 CHULA 29
1815 KLINGON 30
1820 NELLAN 31
1825
----- end of program -----

```

A Visit To The Enchanted House

Nellan is Thirsty

Let's pretend that you are in an enchanted house.

I will say what is happening to you and you should use one or two words to tell me what you want to do. I'll suggest meanings at first.

If you have a question while you're inside the house, type HELP and perhaps I can help you.

When you have read all this, please push the button that says RETURN (or ENTER).

Are you the only person playing (YES or NO)? NO

Please use capital letters.

Are you the only person playing (YES or NO)? NO

How many of you will be playing this time? 2

Please tell me your name

EVERETT Please give me the name of the next player.

TORRY Please give me the name of the next player.

ALLISON Please give me the name of the next player.

ARRENTA Please give me the name of the next player.

B BOV Please give me the name of the next player.

ROAH Please give me the name of the next player.

LAUREN Please give me the name of the next player.

Thank you.



You are in a room called the Bank. A giant carpet on the floor says, WELCOME EVERETT.

A screen says that you are invited to visit for 72 scoots. 0 scoots have passed. To leave this room, type NORTH (or N or GO NORTH).

The BANKER, Mr. KLINGON, looks happy.

There is a magic MAP here. If you GET the MAP, you may later CONSULT the MAP.

==== What do you choose to do?

GET MAP

Oh! you are now carrying the MAP.

The BANKER, Mr. KLINGON, looks glad to see you.

==== What are you going to do now?

GO NORTH

Now you are in the White Room--the walls are white. There is a wall to the west. You know that the Bank is through the south doorway. There is a door to the north and a doorway to the east.

A white RABBIT is here. She hops to you and says,

<< I am CHULA. Welcome to the Enchanted House.

my friend, EVERETT!>>. She hugs you and says,

<< I have a present saved for you in the room to the east of here. If you say TAKE RABBIT (or

EAT CHULA) and then say GO EAST (or the E or EAST), you can carry me to the Gold Room!>>.

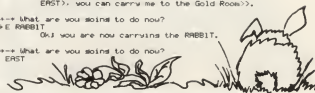
==== What are you going to do now?

TAKE RABBIT

Oh! you are now carrying the RABBIT.

==== What are you going to do now?

GO EAST



All the walls in this room are painted gold. There are exits to the north, east, and west.

CHULA hops out of your arms and scurries off. She returns and says, <<Please TAKE THIS golden EGG that I hid here in the Gold Room!>>.

A golden EGG is on the floor.

==== What now, EVERETT?

TAKE EGG

Congratulations, EVERETT. You'll earn points when the EGG is DEPOSITED (or DROPPED) in the Bank.

CHULA the white RABBIT is here and she says << Well, my friend EVERETT, I hope that you are enjoying your adventure!>>.

==== What is your decision?

DRINK EGG

The Egg is hard to DRINK.

==== What are you going to do now, EVERETT?

CONSULT MAP

You are in the Gold Room now.

==== What do you choose to do, EVERETT?

GO WEST

You are in the White Room—the walls are white. There is a wall to the west. You know that the Bank is through the south doorway. There's a door to the north and a doorway to the east.

==== What is your decision, EVERETT?

GO SOUTH

You are in a room called the Bank. A giant carpet on the floor says: WELCOME EVERETT. A screen says that you are invited to visit for 72 scoots. 4 scoots have passed. To leave this room, type NORTH for N or GO NORTH.

The BANKER, Mr. KLINKOVN, looks glad to see you.

==== What are you going to do now?

DEPOSIT EGG

The BANKER, Mr. KLINKOVN, looks busy. Type SCORE to check your score.

A golden egg is on the floor.

==== What is your decision, EVERETT?

SCORE

You have 25 points out of 100.

You earn points by DEPOSITING valuable things in the Bank.

==== What is your decision, EVERETT?

NORTH

S-S-S It's TOMMY's turn now. S-S-S

You are in the White Room—the walls are white. There is a wall to the west. You know that the Bank is through the south doorway. There's a door to the north and a doorway to the east.

==== What do you choose to do, TOMMY?

WEST

The door is locked and you're not carrying the key.

==== What are you going to do now?

GET KEY

It seems that the KEY isn't here.

==== What now?

GO EAST

All the walls in this room are painted gold. There are exits to the north, east, and west.

CHULA the white RABBIT is here and she says << Well, my friend TOMMY, I hope that you are enjoying your adventure >>.

==== What do you choose to do, TOMMY?

NORTH

This is the Milk Room. If you have a BOWL, you can say FILL BOWL (or GET MILK) and have your BOWL filled with nice cold MILK. There are doorways to the south and east.

An old sun WRAPPER is on the floor.

==== What is your decision?

GET WRAPPER

OK, you are now carrying the WRAPPER.

==== What now, TOMMY?

GET MILK

You need a BOWL.

==== What are you going to do now?

GET BOWL

That's not possible now, TOMMY.

N
|
W---E
|
S



==== What are you going to do now, TOMMY?

GO EAST

You cannot enter—the doorway is blocked by a thick swarm of carrots.

==== What is your decision, TOMMY?

GO SOUTH

All the walls in this room are painted gold. There are exits to the north, east, and west.

CHULA the white RABBIT is here and she says << Well, my friend TOMMY, I hope that you are enjoying your adventure >>.

==== What do you choose to do?

EAST

Oh it's hot in here. You are in the Hot Room and if cold MILK were in this room, it would be warmed. The MAP shows the doorways.

On the floor is a COUPON. According to what is written on the COUPON, if it is DROPPED in the Store while the machine is working, it will be good for one blue BOWL.

A voice says. << Try HELP if you need help >>.

==== What is your decision, TOMMY?

GET COUPON

OK, you are now carrying the COUPON.

==== What are you going to do now?

EAST

You can't walk through walls.

==== What now, TOMMY?

CONSULT MAP

You are in the Hot Room now.

N
|
W---E
|
S

==== What do you choose to do now?
S-S-S It's ALLISON's turn now. S-S-S

==== What are you going to do now?

SOUTH

You are in the room where NELLAN lives! This room is called the Cat room.

NELLAN the CAT is here. She pretty much fills up the room she's as large as an elephant—no kidding. In a soft polite voice she says: << I'd love to have some nice cold MILK. Many of my friends have tried to deliver cold MILK here and many have failed. Perhaps you, ALLISON, would be kind enough to try. I must warn you! It will not be easy >>.

==== What is your decision, ALLISON?

PET CAT

I don't understand PET CAT. Perhaps you would like to type LIST WORDS to see a list of some of the words that I understand.

==== What do you choose to do?

LIST WORDS

I know these verbs: TAKE, GET, GRAB, OBTAIN, DROP, GIVE, OFFER, DEPOSIT, CONSULT, OPEN, UNLOCK, FILL, LIST, NORTH, SOUTH, EAST, WEST, DOOR, CHAIR, RABBIT, BOWL, COUPON, WRAPPER, KEY, EGG, COIN, JEWEL, CATALOG, URGE, MILK, CAT, WORDS, INVENTORY, SCORE, WATCH, CHULA, KLINKOVN, NELLAN.

I know these nouns: N, S, E, W, NORTH, SOUTH, EAST, WEST, DOOR, CHAIR, RABBIT, BOWL, COUPON, WRAPPER, KEY, EGG, COIN, JEWEL, CATALOG, URGE, MILK, CAT, WORDS, INVENTORY, SCORE, WATCH, CHULA, KLINKOVN, NELLAN. I know some one word messages (for example: HELP) and I know some two word messages (for example: FILL BOWL, OFFER MILK). Whatever you say, I'll look at the first and last words: you could say SIT IN THE CHAIR or CONSULT THE SLEEPY MR KLINKOVN.

==== What is your decision, ALLISON?

GET CAT

The CAT is hard to GET.

==== What now?

WEST

You are standing in a room with green walls. A green CHAIR is in the corner. A sign here says:

| Magic works |
| in the |
Enchanted House.

A heavy COIN is here.

==== What are you going to do now?

GET COIN

Congratulations, ALLISON!



You'll earn points when the COIN is DEPOSITED (or DROPPed) in the Bank.

==== What do you choose to do, ALLISON?
CONSULT MAP

You are in the Green Room now.

```

      N
      |
      +---+ E
      |
      S
  
```

```

+++++ Milk +++++
+++-----+++
+++ White Gold Hot +++
+++ Bank Green Cat +++
+++++-----+++++
  
```

==== What do you choose to do?
HELP

Lost? --- CONSULT MAP
Not sure what you're carrying?---INVENTORY
How many scoots are left?---CONSULT WATCH
Want the room redescribed?---RELOOK
What's a certain thing worth?---WORTH that thing
Examples: WORTH COIN, WORTH CATALOG, WORTH CAT

Want a hint?---CONSULT a friend
Examples: CONSULT BANKER, CONSULT CHULA
Not sure how to saw something? --- LIST WORDS
Not sure what the clues mean? --- Think & experiment

==== What is your decision?
WORTH WRAPPER

The WRAPPER is worth 0 points.
The WRAPPER might be useful as a tool.

==== What now?
WORTH CAT

The CAT is worth 0 points.
The CAT might be useful as a tool.
Friends are valuable.

==== What is your decision?
CONSULT WATCH

11 scoots have passed of your visit
& 61 scoots remain.

==== What are you going to do now, ALLISON?
CONSULT CAT

It seems that the CAT isn't here.

==== What are you going to do now, ALLISON?
E

You are in the room where MELLAN lives: this room called the Cat room.

MELLAN the CAT is here. She pretty much fills up the room: she's as large as an elephant--no kidding. In a soft polite voice she says: << I'd like to have some nice cold MILK here. Many of my friends have tried to deliver cold MILK here and many have failed. Perhaps you, ALLISON, would be kind enough to try. I must warn you! It will not be easy >>.

==== What are you going to do now?
CONSULT CAT

The CAT laughs and says: <<Well, I know that you know a carrot-like rabbit>>.

==== What is your decision, ALLISON?
THINK CAT

I don't understand THINK CAT. Perhaps you would like to type LIST WORDS to see a list of some of the words that I understand.

==== What do you choose to do?
N

Oh it's hot in here. You are in the Hot Room and if cold MILK were in this room, it would be warmed. The MAP shows the doorways.

==== What are you going to do now?
U

All the walls in this room are painted gold. There are exits to the north, east, and west.

CHULA the white RABBIT is here and she says << Well, my friend ALLISON, I hope that you are enjoying your adventure>>.

==== What now, ALLISON?
GET CHULA

You can only carry 4 objects (not counting the WATCH) and you are already carrying that much. If you really want to GET something else, then you will have to DROP something first.
To see an inventory of what you're carrying, type INVENTORY.

CHULA the white RABBIT is here and she says << Well, my friend ALLISON, I hope that you are enjoying your adventure>>.

==== What is your decision, ALLISON?
INVENTORY

Your WATCH stays on your wrist for the whole visit. Here is an inventory of what you're carrying:

```

COUPON
MAP
  
```



WRAPPER
COIN

==== What do you choose to do, ALLISON?
DROP WRAPPER

Ok.

CHULA the white RABBIT is here and she says << Well, my friend ALLISON, I hope that you are enjoying your adventure>>.

An old sum WRAPPER is on the floor.

==== What is your decision?
GET CHULA

Oh! you are now carrying the CHULA.

An old sum WRAPPER is on the floor.

==== What now?
N

It's ARIENTA's turn now. ---E---

This is the Milk Room. If you have a BOWL, you can saw FILL BOWL (or GET MILK) and have your BOWL filled with nice cold MILK. There are doorways to the south and east.

CHULA hops down and slobbers a mouthful of carrots that was blocking the east doorway. She smiles and says: << Thanks, friend ARIENTA >>.

==== What are you going to do now?
E

You are in the Store.

In a corner of the room is a machine which announces that if you DEPOSIT (or DROP) a COUPON and the machine is working, you will receive whatever the coupon is good for. If the machine doesn't work, you are invited to call the repairperson.

==== What do you choose to do, ARIENTA?
DEPOSIT COUPON

On the floor is a JEWEL.

There is a clean empty blue BOWL here.

The machine in the corner has a flashing display announcing <<out of BOWLS>>.

On the floor is a JEWEL.

==== What is your decision?
GET BOWL

Oh! you are now carrying the BOWL.

The machine in the corner has a flashing display announcing <<out of BOWLS>>.

On the floor is a JEWEL.

==== What now, ARIENTA?
GRAB JEWEL

Congratulations, ARIENTA. You'll earn points when the JEWEL is DEPOSITED (or DROPPed) in the Bank.

The machine in the corner has a flashing display announcing <<out of BOWLS>>.

==== What now?
GO BACK TO THE WEST

This is the Milk Room. If you have a BOWL, you can saw FILL BOWL (or GET MILK) and have your BOWL filled with nice cold MILK. There are doorways to the south and east.

CHULA the white RABBIT is here and she says << Well, my friend ARIENTA, I hope that you are enjoying your adventure>>.

==== What is your decision?
GET SOME NICE COLD MILK

Guns! slurs! plink. Your BOWL has cold MILK now.

==== What is your decision?
DRINK THE MILK

Guns! slurs.

==== What do you choose to do, ARIENTA?
FILL THE BOWL

Guns! slurs! plink. Your BOWL has cold MILK now.

==== What is your decision, ARIENTA?



Cryptarithms From "Way Back"

BY JOHN DAVENPORT CREHORE



John Davenport Crehore, 1970

CRYPTARITHMS! Fascinating puzzles for fifty years! In what? In math. Oh... yes, *arithmetic* for all—kids to P.G.s! Ten letters for the ten digits; but which are these *two* digits: $R \div R \div R : TR$? Easy.

In the 1970s cryptarithms began to attract serious attention as stepping stones to *greatness* in mind power—yes, in reasoning-power—both deductive and inductive reasoning, vastly different, not generally exercised together in formal education.

In the early 1930s a WWI flying pal of mine joined The American Cryptogram Association and the National Puzzlers' League. In 1936 he sponsored me for membership in both societies. Just now I am in only ACA. Let me introduce my WWI pal, with more to come. He is Col. F. D. Lynch, U.S.A.F. (Ret.) of Florida. He took the fraternal name FIDDLE. I came in as NINE HEX in reference to a block puzzle I had promoted years before.

The ACA was organized September 1, 1929, in Biloxi, Mississippi, to explore the pleasurable scientific qualities of cryptic writing. The group established a bi-monthly magazine, *The Cryptogram*. In time a little department of cryptarithms became one of the regular features. The word cryptarithm was then

colloquial but has finally been accepted and listed in *Webster's New International Dictionary*, 3rd Edition. The word was put together to describe arithmetic in hidden, cryptic form expressed in letters of the alphabet to designate the ten digits 0 to 9. The challenge is to determine each letter's digit value from its position and function in a puzzle.

There are now periodicals offering all sorts of mathematical puzzles: sticklers or just plain, simple, relaxing *fun* in twists that fascinate all ages. Cryptarithms abound. *Recreational Computing* is one of these periodicals. *RC's* introduction stressed "entertainment and home education." Another journal entitled "Cryptologia" is published by Albion College, Albion, Michigan 49224.

The authoritative, perspicacious *Wall Street Journal* published a front page article by Staff Reporter Liz Roman Callesse, on March 13, 1978, headed "A Little Calculating And a Lot of Terror Equal Math Anxiety: Colleges and Others Attempt to Counter the Affliction: Tax Expert Hates Figures." The following is a quote from this article:

"If you're scared to death of mathematics, you are not alone. Listen: I can't do math and I don't want to do math, because it's uncomfortable.

Math anxiety, say a growing number of educators, is a severe ... problem ... afflicts women most ... keeping them from ... male-dominated fields and ... higher-paying jobs."

Students are flocking to colleges that have set up comprehensible elementary math courses. Persons who fear math find themselves handicapped even in situations designed to aid them. Many can't use pocket calculators. Executives impose arithmetic upon subordinates. Today there are millions of people who feel anxious about their scanty ability to do arithmetic. Why are crossword puzzles so popular? Because millions of people feel pleased that they have increased their education every time they find a wanted word. Cryptarithms offer the attractiveness, the fun, of crossword puzzles plus subconscious education in



Col. F.D. Lynch, U.S.A.F. (Ret.), 1970

arithmetic. Real fun in math education is new; cryptarithms are a delight to School Boards; cryptarithm-solving competition is already being played up in periodicals.

When I was admitted to the ACA, it was a great playing field of professional secret code users, from department store price labellers to international spies. Our national conventions, held jointly with our soro-fraternal society, the National Puzzlers' League, have always scintillated with intellect and risibles. I enjoyed greatly the one at Baltimore, Maryland, where I was sailing away in the spelling match until I missed that one word. Y'know?

WWII prevented attendance by those of our cryptanalysts who were in sensitive positions, but anyway we common folk frolicked on the green uh-h-h carpet if not -sward. We staged baffling charades, guessing games, spelling-backwards bees, extemporaneous orations, speed tests in ratiocination.

A little old lady in (Aha! I fooled you!) seven-league boots, had travelled so widely we set her up as question answerer. Her instantaneous recall of facts and events stunned even us know-it-alls. She called herself a storehouse of useless information. We called her a marvel of mental keenness in old age.

One of our conventions, to give itself a taste of old times, went on an excursion on a Mississippi River paddlewheel, wood-burning steamboat.

We have now a real live little old lady who treasures a scroll awarded her by our societies for having solved 12,000 cryptic puzzles! She is Mrs. Dorothy K. Thomas, DOT. Failing eyesight and age (in the eighties?) have stopped her puzzle solving. But DOT says she enjoys her memories, living quietly in a little New Jersey town.

Dear NINE HEX;

I read and solved *most* of your cryptarithms... What intrigues me is your former association with the National Puzzlers' League.

I do believe I recognize your name from the old issues of *The Enigma* (NPL) that I have read, and am surprised and pleased to hear of you again...

NPL...hard times... '60s... early '70s...today thriving... Enclosed...program...The 140th Convention of the National Puzzlers' League, Stamford, Connecticut, July 20-22, 1979. ...names...new...fun...same...

Sincerely,
Will Shortz (Willz)
Associate Editor
Games Magazine

There is certainly a strong appeal to teenagers in cryptarithms and other features of our two societies, ACA and NPL. I know of several who joined at age 13 or 14. One, now a Ph.D. in Computer Science, Dr. Frank Rubin (FIRE-O of ACA), joined at 14 and at 16 submitted an article that the Editor of *The Cryptogram* judged too technical for our membership! A later editor published it in 1975. That article treated solving of long multiplications by examining the end digits of the sub-products. I can get it from Cryptogram files, and expect permission from ACA to republish it in *RC* for our "Genius" solvers.

Dr. Rubin lives now in New York State. FIRE-O says a highlight of his young life was a convention in Washington, D.C., circa 1960. He won second prize in the major contest—among masters! A higher light shone when he was called upon to do technical writing on "di Crittografia." in the publication of a French edition of a world-famous Italian book. Degree Ph.D. in Computer Science did it!

If you ever become known as a multilingual technician, beware of Soviet secret agents in guise of novices asking inno-

cuous questions. Dr. Rubin was pestered for many months by phone calls and letters from ingeniously concealed sources, untraceable, becoming more and more specific about computer cryptography. Dr. Rubin concluded the man sought aid from unwitting amateurs, in breaking American cipher codes that Soviet professional spies hadn't been able to "crack." He believes the man is still at work. Nota bene, CIA, under your new name!

Another prominent member of ACA since 1949 has been U.S. Navy Captain Victor A. Moitoret (nom Je saurais) of New Mexico. He served as vice-president four years and then as president. He authored a notable pamphlet published by ACA: "Cipher Reference List and Index." His reminiscences emphasize that our two cryptic societies have active members in all 50 states and in 23 foreign countries—1974 data.

Capt. Moitoret points out that cryptography is an ideal diversion for persons restricted in movement. Equipment need be no more than pencil and paper, and textbooks for serious students. *Cryptarithms* provide the most fun clue-by-clue, as one solves the letters for the digits they represent in a problem. Here's a one-minute teaser for beginners: $M \div M \div M :: KM$, with only one digit that can represent M, and only one digit that can represent K.

I can invent others just as easy in seconds; harder ones in minutes. Problems whose solving requires genius, take hours to invent. Over the years I've accumulated hundreds of cryptarithms, unsorted although with solutions attached. Kids love 'em! "Oh! K'n I take this home? I want to see if Mother k'n do it!" (Why never Dad?)

Steady, Atlas! Earth is wobbling!

Cryptarithms are being published so widely that their devotees now may find them in all degrees of complexity. *Recreational Computing* features four problems in each issue, graded for Novice, Adept, Genius—and for Computer. *RC* has hundreds more on call, most of them fairly simple, all with clues or hints available for solution in pure logic with no random trial-and-error labor.

The Cryptogram presents a broad range of cryptography in which cryptarithms have only a small portion. But this broad range offers fascinating entertainment and practice to adepts. Je saurais has an ACA Certificate of Achievement citing 6,000 cryptic problems he has solved. In a newspaper article about him there is a reminder that police departments call upon ACA to decode messages from crank killers. Well, Capt. Moitoret, je saurais where to direct the police

if ever they call upon me. I shall refer them to you.

Near the beginning of this essay I introduced one of the brightest lights in our two soro-fraternities, ACA and NPL, retired Colonel Frederick D. Lynch, FIDDLE. He well exemplifies the intellectual heritage that all addicts to puzzling have, particularly puzzlers in mathematics, which calls for dual reasoning power—deductive and inductive.

Fiddle says that cryptarithms stimulate the imagination while training the mind. He was editor of the cryptarithm section of *The Cryptogram* for several years, and at other times contributed articles on cryptography. He observes that computers rob puzzling fans of the subconscious elation that comes from success in mental struggle.

Col Lynch is an honorary vice-president of ACA, the title recognizing his uncountable services to our societies. In 1964 he published an incomparably useful booklet for cryptarithm addicts—"An Approach to Cryptarithms." Its twenty-four 7" x 9" pages present hints, clues, procedures; tables of "everything" in the arithmetic book: units, squares, roots, factors up through decimals, un-, duo-, to quidecimals; detectable sequences; his "Negation Square" (my "Elimination Table" that I nicknamed for young people, "Is-not Chart").

Besides these useful, simple-to-comprehend aids, the booklet shows dozens of illustrations of situations that no one below a twelfth grader would really grasp! Altogether this is a tremendous, encyclopedic cryptarithm-solving reference book. ACA sells it (at least to members) at the nominal price of \$1.50. As minimum postage must be 48¢, I suggest that person wanting it send \$2.00 to Miss Isabel Murdock, Treasurer, The American Cryptogram Association, 1007 Montrose Avenue, Laurel, Maryland 20810.

The Association does not bid for members but would welcome the kind who would "join the Club" in spirit and perhaps action, but would not waste time in mere correspondence. Initiation and first year dues are \$10; current year dues are \$5; three years, \$10.

Now, if "Club" members can forgive the metaphor, let me remind all that arithmetic is attracting attention fast as a "discipline" to play with and not dread. I hope this sketchy tale of persons and their work and play will lure other lively minds to come to the feast—our feast of cryptic arithmetic, CRYPTARITHMS.

Jack Crehore—NINE HEX

trs-80 art?

BY KEN HAHN

Tired of trying to use SET'S, RESET'S or PRINT'S to make your TRS-80 do something speedy on your screen, but they are just too slow? Then here's a fast one for you!

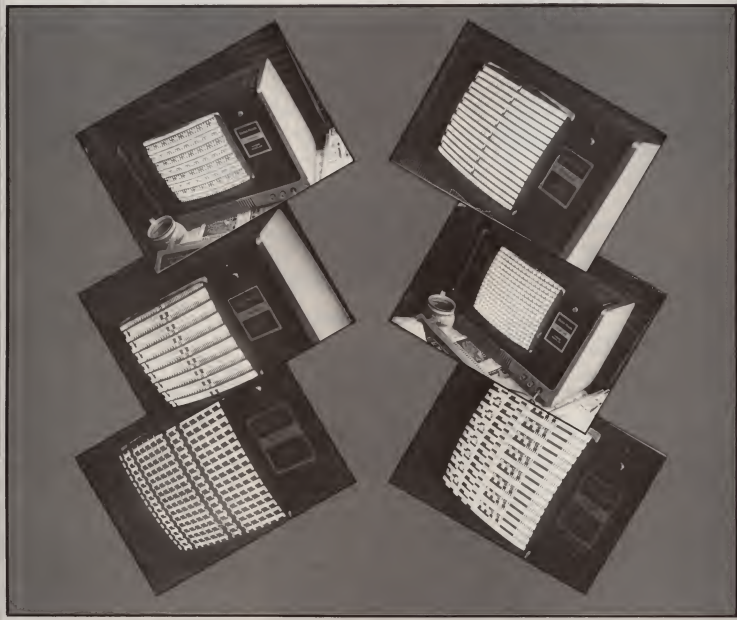
Here's the listing:

```
10 FORZ = 18000 TO 18013
20 READA:POKEZ, A
30 NEXTZ
40 DATA 33, 0, 56, 17, 0, 60, 1, 0, 4
50 DATA 237, 176, 195, 80, 70
60 SYSTEM
```

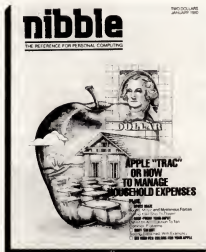
Run this and when a '**?' comes up, type in '/18000' and the screen will fill up immediately. Try pushing in different keys, and the video will reflect the change in micro-seconds.

This program is a short machine language program, and it works by mapping out the keyboard memory on the screen. By changing the byte '56' on line 30, you can map out different portions of your RAM'S and ROM'S on the video.

Experiment by pushing in several keys at once with different combinations. For example, try 'DG349E' and watch it go!



INTRODUCING . . . NIBBLE THE REFERENCE FOR APPLE COMPUTING

**NIBBLE IS:**

A *SOFTWARE GUIDE* for high quality Applications Programs for your Home and Business.

NIBBLE IS:

A *REFERENCE GUIDE* to new Programming Methods.

NIBBLE IS:

A *BUYERS GUIDE* for making purchase decisions on new products.

NIBBLE IS:

A *CONSTRUCTION PROJECT COOKBOOK* for adding function and value to the system you already own.

NIBBLE IS:

A *COMMUNICATIONS CLEARING HOUSE* for users, vendors, and associations.

Each issue of NIBBLE features at least one significant new application program of commercial quality. The programs in NIBBLE are surrounded with articles which show how to USE the programming methods in your OWN programs.

Examples of upcoming articles:

- ☐ Modeling and Forecasting Your Business ☐ Build a Two-Tape Controller for \$12
☐ Arcade Shooting Gallery — Save Your Quarters! ☐ Data Base Management
System I, II, III

And many many more! NIBBLE will literally "Nibble Away" at the mysteries of your system to help you USE IT MORE. In 1980, the principal featured system is the Apple II.

Try a NIBBLE

nibble

BOX 325 Lincoln, Mass. 01773

I'll try NIBBLE !

Enclosed is my \$15 for 8 issues.

☐ **check** ☐ **money order**

Name _____

Address _____

City _____

State _____ Zip _____

COMMENTARIES

MACHINES of LOVING GRACE

- A CELEBRATION

BY DAVID THORNBURG

What started out to be a community workshop on microcomputers turned out to be a stage debut for Big Trak, a mobile toy tank that is also a computer. In fact, Big Trak was the star of the show! — ED

A DAY AT THE COMPUTER ZOO

Several months ago (December 16th, to be exact), many people in the San Francisco Bay Area were doing their holiday shopping. Some people, however, had a much different experience that day—an experience that included Greek dancing, a pot-luck dinner, and the freedom to experiment with a large variety of personal computers. These people attended “Machines of Loving Grace,” a one-day workshop for “dragons and children of all ages.” The workshop, which takes its name from the poem “All Watched Over by Machines of Loving Grace” by Richard Brautigan, was co-sponsored by Dymax (with Bob Albrecht and Ramon Zamora) and by Innovision with the generous support and help of several people, including Liza Loop of the LO*OP Center, Katie Thornburg of the Peninsula School Computer Project, and Dave Offen and Margie Rausch of the Peninsula School.

The workshop took place in the Peninsula School Auditorium—a large room in the old Coleman Mansion in Menlo Park. Tables were set up around the periphery of the room and a variety of computers was brought in: two Apples, two TRS-80's, five PET's, an Atari 800, and two Big Traks. Big Traks?—Ah yes, more on that later!

With the computer set up and tapes of Greek music playing in the background, we were all set for an experience that would be shaped by the desires of those who came. The object of this workshop was to provide the resources and let the

participants create their own activities. After all, since a computer could be turned into a world of caves and goblins, a space cruiser, a lemonade stand, or a splendid word game, our challenge was to provide a wide spectrum of activities for which people could make their selections and let the magic of the environment provide the cohesive structure for the day.

As the activities got started, many of the visitors gravitated towards the computers to sample the wealth of existing software we had on hand. As the clicks and beeps from Petball, SwordQuest and other popular games began to mingle in the air with the music and the laughter of the participants, Albrecht, Zamora and I became aware of another phenomenon taking place in the room. Several of the children (ages five to twelve or so) had discovered the Big Traks and were teaching themselves how to write programs. Big Trak, for the benefit of those of you who haven't seen one, is a large grey tank-like toy, actually a computer, that can be programmed to execute up to sixteen instructions in sequence. These instructions can cause the device to move forwards, backwards, turn left or right by various amounts, pause, and flash its lights. It contains a built-in keyboard connected to a TMS-1000 microprocessor and is programmed in a language similar to LOGO or SMALL-TALK.

Once one child had learned to program this gadget, he would share this knowledge with another child, until a whole crowd had mastered the rudiments of this device. Thus, with almost no adult intervention, several children mastered a computer programming language, learned how to de-bug programs, and were able to use this knowledge to create dances for these machines to perform.

For me, a highlight of this activity came

when one of the younger children ran over, his eyes wide with excitement.

“Dave! Dave! Come see what I did!”

This child had written and debugged a program which caused the Big Trak to traverse the length of the auditorium, turn through a doorway, travel the length of a hallway, turn through another doorway, and go into the foyer of the building—all without bumping into anything. To me this task itself was not nearly as exciting as the fact that this child had learned the key to making a computer do what he wanted it to do. Not bad work for a six-year old!

After dinner, it was finally time to shut the workshop down. One-by-one the computers were turned off and packed up and tables were put away. At last only one table was left with a PET and an Apple, each still getting heavy use. But by this time we had pretty well convinced the remaining enthusiasts that we would do this again soon. With the last table put away, the auditorium was once again quiet and the lights were turned off, bringing a most magical day to a close.

In his poem, Richard Brautigan says:

I like to think (and the sooner the better!) of a cybernetic meadow where mammals and computers live together in mutually programming harmony like pure water touching clear sky.

Well, Richard, for a few hours I think we did it!

(Note: The workshop described above was donated as a benefit for the Peninsula School Computer Project. Future workshops will benefit other projects as well, including Computertown, U.S.A.)



IN DEFENSE OF HACKERS

BY STEVE KUDLAK

Big Trak is a programmable toy vehicle created by the Milton Bradley Company. A child (or an adult, for that matter) can program it to follow a prescribed path. Big Trak has a total of 16 programming steps which direct the motion of the vehicle. The user need only estimate distances and punch in his commands to direct the tank around chair, under a table, and back. Macro commands such as *repeat* and *reverse* and an edit feature add to the toy's versatility. Along the way Big Trak can use its sophisticated defense system to protect itself by firing its sound-and-light laser cannon in single shots or in bursts (either long or short). Milton Bradley sees Big Trak as "a toy that stimulates young imaginations while helping children to prepare for the sophistication of that great big computerized world out there. One 9V and four "D" batteries are all that's required." Approximate retail price is \$43.00.

you can tell
it want to Do
but you have
to tell it just
right
this toy is
Fun cuz it
has a calculaton
on it and I
like it
bigtrak
bye Lisa

Big Trak review
by Lisa Albers
Age 6

In one of my rarer optimistic moments I decided to produce my essay (as in the French word meaning "to try") in defense of hacking. I hope not to be too optimistic or ridiculous. We might entitle this one of the seven weird ideas of SKU; I don't know if it is worth distribution to large mailing lists, but I shall take my chances.

Weird idea number one: "Why is hacking so popular?" I feel that hacking is one of the few areas in modern life that comes close to the idea behind certain Renaissance guilds. For example, in general when people start hacking they learn something themselves, and when they encounter trouble they find an "adept" in the particular system or program and make inquiries getting information that is impossible to find in manuals or hard to find in documentation. During the course of this exchange, many working relationships and some personal relationships are formed.

The second guild aspect of hacking is that once you learn a very small set of things, you can DO something. It isn't like having a single person tie up several million pieces of equipment which he very well might damage. Have you ever tried to load an ultracentrifuge at 2 A.M. and not know if you are really balancing it right? If you do something wrong it gets wrecked, not just - oh well, my program died . . .

The other interesting thing I have seen is the WIDE variety of people communicating with one another. It is truly amazing. People who normally would not consider talking to each other end up conversing freely and even becoming friends. The other interesting aspect of the communication is that it causes a social levelling. I have seen Army colonel types learning from people twenty years their junior and barely out of high school.

The thing I find most encouraging about hackers is that on the whole the lot are COOPERATIVE rather than competitive. Maybe this is the result of being in a "protected" environment, but somehow I feel that even if this is true, the

cooperation is nonetheless valuable and should be encouraged. In a world where people can usually only be gathered together in cases of massive crisis or competitive games, this seems reassuring. The idea that another person wants to see your program succeed and is willing to help is pleasant and good.

In general some of the criticism of hackers being a narrow subculture is valid; however, I don't think the situation is worse than in other fields. For example, a hacker might worry about why the FOO JSYS returns something in AC2 or assumes everyone know that a AOBJN and a CAMLE are two different things operationally. This person is certainly no worse off in my mind than a person who worries about his "peets" and "tris" and "traps" and persists in struggling against the force of gravity to lift larger and larger metal objects into the air. Even funnier is that these people expect admiration and they often get it. Perhaps there are benefits from weight lifting, but certainly it is questionable whether lifting 70 Kg in the air is inherently any better than learning things to do with the CRJOB JSYS. (Please don't try to convince me that lifting metal objects in the air is really wonderful . . . I had a girlfriend who tried this for months. I could appreciate how she liked to do it, but I couldn't see anything wonderful in it myself.)

If there is one thing to be said about Americans it is that they LOVE to build, tinker and play around with devices. I think playing with technology is a good thing, and playing with technology in a supportive and cooperative environment is an even better idea. By playing with technology you can discover new uses for it.

The point is that technology can be a useful thing if people know its real as opposed to its advertised potential. Understanding it also makes it a LOT LESS FRIGHTENING. And loads of people get frightened by something they don't understand or haven't experienced.

In my opinion, people who know the potential of technology should try to increase technological literacy on a wide scale.

I don't think a technologically literate society is impossible. There are places where as much as 20% of the population is technologically literate. It can probably be achieved, provided we don't blow ourselves off the earth with a variety of political/nationalistic stupidities. The world would be much better off because of this enrichment.

TOUCH PANELS AND INTERACTIVE GRAPHICS

BY JIM DAY

A touch panel is a device that senses the location at which it is touched by an object such as a fingertip. A number of schemes have been used to implement touch panels. Some use resistive membranes and some use light beams. The latter seem to be the most commonly used.

Typically, infrared emitting diodes and infrared sensing diodes are paired in a rectangular array mounted in front of a CRT screen. Electronic circuitry is used to sense which horizontal and vertical light beams are broken, passing this information to an associated computer. A computer program translates these data into X and Y coordinates of the touched location. This, in turn, allows applications programs to relate the touch of a finger (toe, nose, or whatever) to graphic or textual information being displayed on a specific area of the CRT.

This sounds simple enough and, in fact, it is simple and easily done. The combination of a touch panel and a CRT provides a very powerful tool for a wide variety of applications when used with interactive graphic displays. For example, instead of using a conventional electro-mechanical keyboard, one can simply display a picture of a keyboard on the CRT and input the keystrokes via the touch panel.

But using a touch panel to simulate a keyboard is a bit like designing an automobile in the shape of a horse. It can and has been done, but there are better ways of using a touch panel. Properly used, a touch panel provides a much friendlier interface than a keyboard. People who are intimidated by keyboards or feel that they will lose face by using a keyboard (traditionally a menial clerical function) are happy to use a touch panel. One example of how a touch panel can be used is in conjunction with a chess program. Instead of typing "D2-D4", one can simply touch the image of the man to be moved and then the image of the location to which it is to be moved. Touch panels can also be used with many other games such as Othello or checkers.

Touch panels are ideal for monitoring and controlling complex industrial processes such as oil refineries, assembly lines, and power plants. Instead of working with huge control panels with hundreds of buttons and flashing lights, one can use a single touch panel and CRT. Only those buttons actually needed at the moment need be displayed. Other buttons, text, and graphic indicators can be displayed as operating conditions change.

The same sort of thing can be done by the use of a "mouse" or a light pen. But a touch panel is a more convenient and natural interface. A computer equipped with a touch panel and CRT is ideal for a comprehensive residential command and control system. Various devices such as radios, television sets, telephones, and heating, cooling, and security circuits could be tied into the computer and controlled via the touch panel.

In the initial or "standby" mode the CRT would be blank, to avoid burning the screen with a constant picture. Or a constantly moving message, such as "Touch to view menu", could be displayed. A touch anywhere on the screen would cause a menu of primary functions to be presented (see Figure 1).

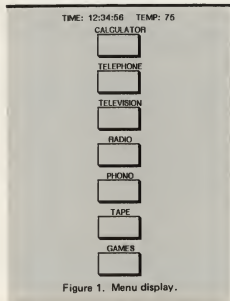


Figure 1. Menu display.

In the "menu" mode, the display could show the time and temperature and perhaps the current stock market report from a service such as Digicast. Below this display area would be a menu of touch-sensitive buttons for selection of various subsystems. The CALCULATOR button, for example, could display the buttons and standard readout of a desk calculator (see Figure 2). All of the buttons would be fully functional, thanks to the touch panel and, of course, the associated computer. Touching the MENU button would cause a return to the menu display.

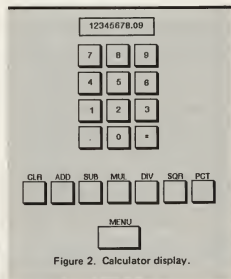


Figure 2. Calculator display.

Touching a TELEPHONE button on the menu could display a touch-tone keyboard, and a scrolling window above the keyboard could show a list of frequently called numbers (see Figure 3). An AUTO-DIAL button could dial a number selected by scrolling the display, or the touch-tone buttons could be used instead. During a telephone call, a videophone picture of the person called could be displayed on the CRT, assuming that such a service is available. At the bottom of the screen, a superimposed GOODBYE button could be used to end a call, or the computer could be programmed to respond to the spoken word "goodbye". A PLAYBACK button could be used to review calls made in, say, the last 24 hours, including those recorded while no one was at home.

Annie	123-4567	UP
Bevle	234-5678	DN
Clyde	345-6789	

1	2	3
4	5	6
7	8	9
*	0	#

AUTO-DIAL

PLAYBACK

GOODBYE

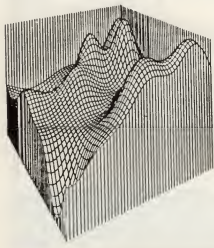
MENU

Figure 3. Telephone display.

Touching the TELEVISION button on the menu could display a monochrome or color TV picture with superimposed buttons and indicators for channel selection, fine tuning, etc. A better scheme would be to show the control buttons only if any part of the touch panel is touched. Otherwise the TV picture would fill the entire screen.

Likewise, a RADIO button could display all of the buttons and indicators required to operate an AM/FM radio, and PHONO and TAPE buttons could display controls appropriate to phonographs and tape decks.

The list of things that could be controlled by a computer and a touch panel is really endless, and I am sure that we are going to see many fascinating applications in the very near future.



CLUB & CLASSROOM DISCOUNTS.

PEOPLE'S COMPUTER COMPANY PUBLICATIONS

Recreational Computing
Dr. Dobb's Journal

Clubs, users groups, schools and any others interested may take advantage of discounts for quantity orders of each of our magazines. See terms below.

All orders will be shipped via UPS—faster than the post office!

DISCOUNT SCHEDULE (one year)

QUANTITY	DISCOUNT	RC	DDJ
Regular Subscription Rate/Year	none	\$12.00	\$19.00
5-10 copies	10%	10.80	17.10
11-24 copies	15%	10.20	16.15
25-49 copies	20%	9.60	15.20
50-74 copies	25%	9.00	14.25
75 + copies	30%	8.40	13.30

TERMS

- Good in the U.S. only.
- Quantity orders must be made up of one title only—no mixing among titles.
- All copies must be shipped to same name and street address (no P. O. box).
- Payment must accompany order—check or money order only.
- Minimum one-year subscription.

Reader Service

MOOVING?

IMPORTANT

Please advise us of address changes 60 days in advance and attach your magazine mailing label here. Please clip this notice and mail, or send us a reasonable facsimile.



Name (please print new information) _____

Address _____

City _____

State _____

Zip Code _____

Country _____

**Recreational
COMPUTING**



P.O. Box E, 1263 El Camino, Menlo Park, CA 94025

ANNOUNCEMENTS

THIS PUBLICATION IS AVAILABLE IN MICROFORM



Please send me additional information.

University Microfilms International

300 North Zeeb Road
Dept. P.R.
Ann Arbor, MI 48106
U.S.A.

18 Bedford Row
Dept. P.R.
London, WC1R 4EJ
England

Name _____
Institution _____
Street _____
City _____
State _____ Zip _____

ADVERTISING SPACE AVAILABLE



¼, ½, Full Page
Inside
Front & Back
Covers

PLEASE SEND
FOR RATE CARD

Advertising Manager
People's Computer Company
1263 El Camino Real, Box E
Menlo Park, CA 94025
(415) 323-3111

Systems

The Apple III made its debut at NCC. It is a fully integrated computer system with built-in floppy disk drive, up to 128K bytes of memory, color and black & white video. The keyboard is new. It features a standard typewriter layout, a 13-key numeric pad, two user-programmable function keys, full upper-lower case character set (with a shift-lock that does the right thing for software that wants upper-case only), and auto-repeat on all keys. The system display features 80 columns.

Many previously optional features are now standard. The built-in floppy controller can handle up to three additional disks; the drives may be installed by plugging them into a connector provided. A printer port and an RS-232C serial port are standard. The system has a built-in clock and calendar maintained by their own internal battery to maintain the correct date and time even when the computer is turned off. A built-in 6-bit d/a converter can be used to synthesize simple music and voice.

The display includes both a text and graphic modes. Character matrices are stored in RAM so fonts may be reconfigured. In addition to 80-character text, a 40-character color-on-text mode and a 40-character Apple II emulation mode are provided. A black and white graphics mode (ultra-high resolution) provides a 560 x 192 grid. High resolution graphics with up to 16 colors is possible on a 280 x 192 grid.

With the system comes a new operating system, Pascal, BASIC, and FORTAN. The operating system provides a uniform interface for all programs and makes the PEEK's and POKE's needed for many microcomputer systems unnecessary.

Onyx Systems has Z8000 Unix in a single package with an integral 8-inch Winchester disk capable of running up to 8 users. With a single-unit price of \$18,000 this package has to be one of the best computer buys today for the serious computer user. Contact Onyx Systems, 73 E. Trimbur Road, San Jose, CA 95131, (408) 946-6330.

The PMC-80 is a plug-compatible TRS-80 replacement. Through an exclusive marketing agreement with a Hong Kong manufacturer, Personal Micro Computers, Inc. has begun marketing a plug-compatible version of the TRS-80 Model I, Level II. There are some enhancements: the cassette and power supply are built-in rather than separate units, the cassette interface is simpler and more robust, a built-in monitor allows a standard TV set to be used for the display as well as a TV moni-

tor. Even more interesting, in addition to the usual TRS-80 interface, a 50-pin connector allows interface to the S-100 bus. The PMC-80 is sold by mail order only. A system including cassette tape, 16K memory, Level II Microsoft BASIC in ROM, power supply, computer and keyboard (but excluding the TV monitor) was introduced at \$495; prices as of this writing are yet to be determined but will be competitive with the TRS-80. Contact Personal Micro Computers, 475 Ellis Street, Mountain View, CA 94043, (415) 968-1604.

The IMAGINATION MACHINE from APF Electronics is a new low-cost personal computer for home entertainment. It features user programmability (BASIC), 14K of ROM, 9K of RAM, typewriter style keyboard, two game-style controllers, a 32 by 16 line display format, color, a built-in cassette deck, microphone jack, and facilities to add external peripherals including a printer, telephone modem, mini-floppy, additional memory, and so forth. Graphics resolution is 128 x 192 with up to 8 colors. The suggested list price is \$599. For additional information contact APF Electronics, 444 Madison Avenue, New York, NY 10022, (800) 223-1264.

Software

Stock Market Charting for the Apple II is possible with a \$99.95 technical analysis package available from RTR Software, P.O. Box 12351, El Paso, TX, (915) 544-4397.

The FICOMP Ham DX Package is a package of machine language programs and data which operates under 8080/Z80 CPU systems to support the active radio amateur. The program computes specialized data (bearing, distance, time difference, etc.) from your site to a variety of foreign sites. Cost is \$22 postpaid from FICOMP, 3017 Talking Rock Drive, Fairfax, VA 22031.

A Structured BASIC Translator available from Acorn Software Products, 643 North Carolina Ave., S.E., Washington, D.C. 20003, (202) 544-4259, allows the user to write programs with structured control constructs such as PROCEDURE, CALL, CASE, IF-THEN-ELSE, WHILE, and UNTIL, and then translate the resultant program into standard BASIC. The program costs \$29.95 on diskette or on the Model I, Level II TRS-80.

Pascal/M Release 2 is now available from Sorcim for \$175. The system consists of a compiler, emulator, runtime library and user manual. It operates under CP/M and supports full Pascal plus extensions (including ADA

syntax for random I/O). It will run on any 8080/Z80 system with or without AMD9511 arithmetic processors. An 8087/8088 version is soon to be released. Contact Sorcim, 2273 Calle De Luna, Santa Clara, CA 95053, (408) 988-3963.

Hardware

Cromemco has a high-resolution graphics interface for use with Cromemco computer systems. The video board is fully resolution compatible and may be used as a colorizer or color modulator in a television broadcast studio since it can be synchronized to external television equipment. A variety of different color and resolution options are available providing up to a maximum of 75 by 484 points. It does require special dual-ported memory between 12 and 48K bytes simply for the display. Contact Cromemco Inc., 280 Bernardo Avenue, Mountain View, CA 94043, (415) 964-7400 for more information.

Other

Dr. Dobb's Journal Volumes 1, 2, & 3 are now available in book form from Hayden as part of the People's Computer Company series. Contact Hayden Publishing Company, 50 Essex Street, Rochelle Park, NJ 07662.

All About Personal Computers is a 62-page report from Datapro covering 15 of the best-selling personal computers and including directories listing vendors of hardware, software, peripherals and publications. The price is \$25 from Datapro Research Corporation, 1805 Underwood Blvd., Delran, NJ 08075, (609) 764-0100.

Games Unlimited 1980 is a trade, consumer, and entertainment show scheduled for CT in October in October 23-26. A variety of exhibits, lectures, and seminars are planned. Contact Arnett & Associates, 25 Canton Road, Simsbury CT 06092, (203) 651-8631.

International Apple Core is a nonprofit independent organization which will act as the parent organization for local Apple groups. No individual memberships are available though individuals may subscribe to the IAC quarterly. Apple user groups should contact International Apple Core, Box 976, Daly City, CA 94017 for information.

Apple for the Teacher is a user group specializing in educational uses of Apple Computers. Contact Apple for the Teacher, Ted Perry, 5848 Riddick Street, Citrus Heights, CA 95610 for information.

THE LEAST EXPENSIVE PROGRAMS YOU CAN BUY.

Up to 77 high-quality programs for
TRS-80, only \$8!

THE \$15.50 COMPILER:

TINY PASCAL FOR TRS-80

At last your TRS-80 can run Pascal too! The Chung/Yuen "tiny" Pascal is fully implemented for Level II TRS-80, 16K and up. You no longer need be left out of the growing group of Pascal users, because People's Pascal gives you everything you need to write structured Pascal programs:

- tiny Pascal compiler • complete text editor for writing your programs • complete tiny Pascal monitor • sample Pascal programs • user's manual (TRS-80 Computing Issue 1:4)

People's Pascal is both a powerful, structured language and "CPU expeditor". People's Pascal programs execute at least four times faster than Basic, and often eight-times faster! Special functions open up the complete graphic capability of TRS-80. You now have the means to write those dazzling, impressive, high-speed graphics programs that are great for games, plotting, statistics, etc.

For the serious computerist, side two of People's Pascal II (tape 6) contains a larger compiler and complete source to the compiler, written in Pascal! This means you can re-compile the compiler, making changes, adding features, etc. (but this will take at least 36K RAM and a solid knowledge of programming).

With the complete People's Pascal operating system, you can save and load both source (Pascal) programs, and compiled programs, to or from cassette tape. This means that once you have de-bugged a program, you can save the P-code (compiled program) and thereafter, to run the program, you need only load the super-fast P-code.

Here is a partial list of People's Pascal features: recursive procedure/functions • for (loop) • case if/then/else • one-dimensional arrays • write • read constant • repeat/until (loop) • "peek & poke" • plot (graphics for TRS-80)

DEALER INQUIRIES INVITED

People's Pascal I (tape 3) is written in Basic, implemented for TRS-80 by John Alexander of Berwick Australia. It compiles P codes more slowly and is harder to use than Pascal 2, but its P codes can be translated into Z80 native code and saved as System tapes. Pascal 2 requires that Pascal be resident at run time—Pascal 1 does not. Other People's Software tapes \$8.

—CA Residents Add 6 per cent TAX (\$8.45, \$16.40 & 24.88). Quoted Prices INCLUDE \$.50 Shipping—

TAPE 1 LEVEL 2

Mortgage calculations, Dow Jones Industrial, cash flow, inventory, change, California income tax, journal ledger (8K), loan amortization, perpetual calendar, biorhythm, payroll, diet planning, spreadsheet, touch typing, sales receipt tally, decisionmaker, mail addressing, straight depreciation, double-declining depreciation, and revolving charge account.

Also, math problems, queen, Star Trek I, number guessing, wheel of fortune, World War II bomber, rock-scissors-paper, saak, Star Trek II, Red Baron, mini-Trek, strategy, pilot, battleship, "On A Snowy Evening", mastermind, tic-tac-toe, grand prix auto race, capitals, elch sketch, hangman. Total programs: 34; Level 1 version available: 24 programs. \$8

TAPE 2 Some Common Basic Programs (lev. 2)

Fully documented in Some Common Basic Programs by Lon Poole & Mary Borchers (Osborne & Associates, 630 Bancroft way, Berkeley CA 94710—or from CIE—\$12.50 postpaid from CIE, via UPS, CA residents add tax (to \$13.25)).

Investment, future value regular deposits; regular deposits; regular withdrawals, initial, minimum (for withdrawals); nominal interest, effective & earned interest; depreciation rate, amount depreciation; salvage value; discount com'l paper; loan principal, regular & last payment, remaining balance, term-loan; mortgage amortization; greatest common denom. - integer prime factors; polygon area, triangle parts; analysis, operations two vectors; radii, degree, arc length, radian conversion; coordinate, polar equation, functions plot; linear, curvilinear interpolation; Simpson's & trapezoidal rules, Gaussian quadrature integration; derivative.

Side 2—quadratic equation, polynomial (Newton) & half-interval-search roots; trig polynomial; simultaneous equations; linear programming; matrix addition, subtraction; scalar multiplication, inversion; permutations & combinations; Mann-Whitney U test; mean, variance, standard deviation; geometric mean & deviation; binomial, Poisson, normal, Chi-square distribution; Chi-sq., student's T-distribution test; F-distribution; linear correlation coefficient; linear, multiple-linear, Nth order, geometric, exponential regression; system reliability, future projections; Federal withholding taxes; tax depreciation schedule; check writer; recipe cost; map check; day of week; days between two dates; anglo to metric; ephemeris. \$8

TAPE 3 People's Pascal Development System 1

Pascal 1 compiler-program development system. \$15.50

TAPE 4 LEVEL 1

Election returns, business percentage, ups and downs of business, index, inventory control, sales receipt tally, gas mileage, driving distance, mixed monthly sales report, payroll, annual earnings, speech recording aid, and double-declining depreciation.

Also, math problems, cash register, chase, snoopy, commander-in-chief, Christmas graphic, air raid, balance scale, stock market, tic-tac-toe and On A Snowy Evening. \$8

TAPE 5 LEVEL 2

Memory test, mortgage payments, tension breaker, lineprinter-screen & vice-versa utilities, Federal income tax, election returns, business percentage, vacation planner, car pool(disk), diet planning 2, mailing list(disk) and first aid.

Also spelling bee, Star Trek 3, mind bender, technoscope, chase, common factor, klugeon capture, spelling practice, Hamurabi, animals, Snoopy, cryptograph, starship, ants, Yesterday, and Pilot(disk). Pilot is the language of computer-aided instruction (CAI).

TAPE 6 People's Pascal 2

Pascal 2 compiler-program development system. \$23.50

TAPE 7 LEVEL 2

Disassembler, Pilot, roster, dropout, memory loader, memory sort, inventory control, graph, land surveying, mixed monthly sales report, shopping list, diet planning 3, loan progress chart, hex-decimal conversion.

Also Star Trek 4, steles and capitals, battleships 2, spelling practice 2, number guessing, hangman 2, snark, slot machine, cipher, target, surround, add, termite, lunar lander, multiplication exercise, five-in-a-row, Bastam, and write. A number after a program indicates there are other similar People's Software programs. Pilot is the same as the disk pilot on tape 5, except runs on 16K tape systems. \$8

COMPUTER INFORMATION EXCHANGE
Box 159, San Luis Rey CA 92068 (714) 757-4849

A Proposed Graphics Language

BY JIM DAY

The display of high-resolution graphics is possible on many kinds of small computers. This is often done by adding special graphics commands to some popular language such as Basic. Graphics commands are generally rather primitive, allowing one to plot points, draw straight lines, and display predefined patterns. If an interpreter is used rather than a compiler, the speed of execution is usually too slow for satisfactory animation.

Because it takes time to change a graphic display, it's a good idea to alternate between two display areas in memory, so the old picture will remain on the screen until the new picture is ready for display. Another way of hiding display updates from the viewer is to blank the screen until the new picture is ready. This may produce flicker in the display but has the advantage of freeing the computer from the task of maintaining the display on the screen, which can easily consume more than half of the computer's time.

Suppose we had a really good graphics language at our disposal. Let's call it Really Advanced Pictorial Image Display, or RAPID for short. Let's assume that this is an interactive language that allows pictures to be defined manually via a mouse, stylus, or joystick as well as by source code. Let's further assume that the language processor is smart enough to translate a manually entered picture into the corresponding source code. It should also be possible to edit a picture online and have the processor modify the source code accordingly.

Ok, so what kind of graphics statements do we want RAPID to understand? To define, but not display, a dot at screen location X,Y we would like to write:

DOT FOO IS X,Y

FOO is the symbolic name assigned to the dot. To actually display the dot after it has been defined, we will use the statement:

DRAW FOO

In real life, one seldom wants to display a single isolated dot. Usually one deals with lines and groups of lines. So we need a line defining instruction:

LINE ZAP IS X1,Y1 TO X2,Y2

This defines a straight line, named ZAP, running from screen location X1,Y1 to location X2,Y2. Naturally, we expect RAPID to complain if either endpoint lies off the screen. If the display device uses raster scanning, most lines won't be perfectly straight but will be approximated by a series of line segments that are either horizontal or vertical. But let's not worry about that. To relocate line ZAP, we can write:

SHIFT ZAP BY X,Y

X and Y specify horizontal and vertical bias values for line ZAP. The length and direction of the line are unchanged. To rotate line ZAP about some point, not necessarily on the line itself, we can write:

ROTATE ZAP ABOUT X,Y BY A

X,Y defines the point about which ZAP is to be rotated and A is the angle of rotation, expressed in degrees. Rotation will be counterclockwise for positive values of A. Nothing happens on the screen until a DRAW instruction is executed:

DRAW ZAP

It would be nice to define a group of lines forming some figure such as a rectangle:

FIGURE FIG1 IS LIN1,LIN2,LIN3,LIN4

The arguments LIN1, LIN2, LIN3, and LIN4 are the names of previously defined lines. FIG1 is the name given to the figure comprising the four lines. We can then use FIG1 in other statements such as:

ROTATE FIG1 ABOUT X,Y BY A

An obvious extension of this is to allow figures to be composed of previously defined figures and/or lines:

FIGURE FIG2 IS FIG1,LIN5,LIN6

FIG2 is composed of FIG1 plus lines LIN5 and LIN6. To duplicate the shape of an existing figure, we can write:

FIG9 IS FIG7

The location of FIG9 can then be adjusted via a SHIFT or ROTATE statement. It would be useful to define circles:

CIRCLE C1 IS R AT X,Y

C1 is defined as a circle of radius R whose center is at location X,Y. Circular arcs are also useful:

ARC ZOT IS R AT X,Y FROM A1 TO A2

ZOT is a circular arc of radius R whose center is at X,Y. It extends from angle A1 to angle A2. To remove graphic elements from the screen we need a statement to do the inverse of what DRAW does:

ERASE FIG3,LIN4

This will erase all elements of FIG3 as well as LIN4. An ERASE statement with no arguments will erase the entire screen. If we want to make an element larger or smaller without changing the shape, we can use a SCALE statement:

SCALE FIG7 BY 2

This will double the size of FIG7, using the center of the figure as a fixed reference point. There should be a way of defining text strings:

TEXT TXT1 IS "FRONT VIEW" AT X,Y

This defines a text element named TXT1 comprising the string "FRONT VIEW" and having a starting location of X,Y. Once defined, a text element can be displayed via a DRAW statement or can be included as part of a figure:

FIGURE FIG7 IS FIG6,TXT1

If a figure is defined as including text, the text can be shifted, scaled, or rotated along with the rest of the figure:

SHIFT FIG7 BY X,Y

Animation can be accomplished by program loops containing appropriate DRAW, ERASE, and element redefinition statements.

BACK ISSUES

PCC NEWSPAPERS

Half Off Sale!

All 12 available back issues of PEOPLE'S COMPUTER COMPANY NEWSPAPER for only \$12.

- #3 May-June 1973 ZINGO/"Going to Bid"/How to Buy an Edusystem/BASIC MUSIC/CHOMP. \$2.00
- #10 July-Aug 1974 A Fantasy of Future Forms/Low-Cost Home School Microprocessor Systems/INCHWORM/MAZE/Analysis of Teaser/Computer Illiteracy Problem by Peter Lykos. \$2.00
- #13 Jan-Feb 1975 Build your own BASIC/Mysterious RND/More INCHWORM/Computers & Art/BIOSIN Son of Hardware. \$2.00
- #18 Nov-Dec 1975 NUMBER/Tiny BASIC/Analog to Digital Conversion/Minuet in G Major. \$2.00
- #19 Jan-Feb 1976 Space Games/Tiny BASIC Soloworks/DIDDLE/STTR/More Music. \$2.00
- #20 Mar-Apr 1976 Games for the Pocket Calculator/SQUARES/Make Believe Computers/STRTRK/Huntington Computer Project. \$2.00
- #21 May-June 1976 Biofeedback/POUNCE/"We're Building Our Own Computer"/Programmer's Toolbox/CCC/TV as Terminal and Game Center. \$2.00
- #22 July-Aug 1976 Low Cost Software/Tiny BASIC, Tiny Trek/Your Brain is a Hologram. \$2.00
- #23 Sept-Oct 1976 Dungeons & Dragons/HATS/One on One/PLANETS/The Positive of Power Thinking. \$2.00
- #24 Nov-Dec 1976 STORY/SNAKE/More Build Your Own Computers/Introducing PILOT/FROGS. \$2.00
- #25 Jan-Feb 1977 REVERSE/Robots/Tiny PILOT/Space & Computers/Conversational Programming. \$2.00
- #26 Mar-Apr 1977 Z-80 PILOT/6502 Assembly Programming/Tiny BASIC for Beginners. \$2.00

Only the issues listed above are available. Price includes issue, handling, and shipment by second class or foreign surface mail. Within the U.S., please allow 6-9 weeks to process your order second class. Outside the U.S., surface mail can take 2-4 months. For faster service within the U.S. we'll ship UPS if you add \$1.00 for 1-2 issues and \$.50 for each issue thereafter. We need a street address, not a PO Box. Outside the U.S., add \$1.50 per issue requested for airmail shipment.

PEOPLE'S COMPUTERS

- #27 May-June 1977 Graphics, Computers & Copyright Laws, PILOT & Tiny BASIC, Home Computers for Beginners, Women & Computers. \$2.50
- #28 July-August 1977 Heathkit's 8080 & LSI-11, Do-It-Yourself CAI, Robot PETs. \$2.50
- #29 Sept-Oct 1977 Computer Networks, The \$595 PET, More Tiny Languages. \$2.50
- #30 Nov-Dec 1977 Oz Graphics, Bio-feedback & Micros, Our PET's First Steps. \$2.50
- #31 Jan-Feb 1978 Robots, Video Discs, PET Update & Programs, PASCAL & COMAL. \$2.50
- #32 March-Apr 1978 Epic Computer Games, TRS-80 Review, Micros for the Handicapped. \$2.50
- #34 July-Aug 1978 Computer Whiz Kids, Public Access to Computers, Man-made Minds. \$2.50
- #35 Sept-Oct 1978 Computer & Museums, Kingdom, APL, Sorcerer of Exidy. \$2.50
- #36 Nov-Dec 1978 The Return of the Dragons, APL, Animated Games for TRS-80. \$3.00

RECREATIONAL COMPUTING

- #37 Jan-Feb 1979 Artificial Intelligence, A Fantasy by Jules Verne, TRS-80 Personal Software, The Apple Corps is with Us. \$3.00
- #38 March-Apr 1979 Artificial Intelligence, Calculator Comics, Chess Reconsidered, Lord of the Rings. \$3.00
- #39 May-June 1979 Now Peter Can Read, Game of Life, TRS-80 Programs, Detective Game Program. \$3.00
- #40 July-Aug 1979 Computertown, U.S.A.I., Cryptarithms, DP Games Playing, FORTE Music Programming Language. \$3.00
- #42 Nov-Dec 1979 Shogi Games, Creating New Characters on the TI 99/4, Music for Atari, TRS-80 & PET. \$3.00
- #43 Jan-Feb 1980 Computing for Health and Equality (PET), Ti Graphics & Animation Pt. 2, More Music for Atari. \$3.00
- #44 Mar-Apr 1980 Special Games Issue! Recreation Apple II Hi-Res Graphics, Galaxy II, TRS-80 Tiny Trek. \$3.00
- #45 May-June 1980 Intro to Computer Music, CBBS Phone Numbers, 6502 Machine Language, Five New Personal Computers. \$3.00

Circle your choices.

3 10 13 18 19 20 21 22 23 24 25 26 27 28 29 30
31 32 34 35 36 37 38 39 40 42 43 44 45

Please send me the issues circled.

☐ Yes, send me #3-26 for only \$12

☐ I'm ordering \$20 or more and I want a 10% discount.

I enclose \$ _____ (U.S. check or money order). Foreign add \$5.00 per issue (this includes postage and handling).

Please send in this form or facsimile to: Recreational Computing, 1263 El Camino Real, Box E, Menlo Park CA 94025.

My Name _____

Address _____

City/State/Zip _____



PROGRAMMING PROBLEMS

© 1980, BY BOB ALBRECHT, DON ALBERS AND JIM CONLAN

PROBLEM #19 WORD'S WORTH NUMBER TWO

You did do PROBLEM #11 WORD'S WORTH NUMBER ONE, didn't you? Of course you did. Oh... you didn't? Well, never mind. If you need it to help you in this one, skip back and do it.

Remember: A = 1, B = 2, C = 3, and so on up to Z = 26. The worth of a word is the sum of the number values for the letters in the word.

Here is a list of the first 100 prime numbers.

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541

For each prime number, find a word whose word's worth is equal to the prime number. We will accept any single unabbreviated word in *Webster's Third New International Dictionary* (Unabridged). Good hunting!

PROBLEM #20 SUM AND PRODUCT OF DIGITS OF A TWO-DIGIT NUMBER

Write a program to compute and print or display the sum of the digits and the product of the digits of a two-digit positive integer. A two-digit positive integer is:

- (1) an integer,
- (2) greater than or equal to 10, $10 \leq X \leq 99$,
- (3) less than or equal to 99.

Sample RUN.

TWO-DIGIT POSITIVE INTEGER? 37
SUM OF DIGITS = 10
PRODUCT OF DIGITS = 21

TWO-DIGIT POSITIVE INTEGER? -37
OOPS! ENTER AN INTEGER, 10 TO 99.

TWO-DIGIT POSITIVE INTEGER? 7.3
OOPS! ENTER AN INTEGER, 10 TO 99.

TWO-DIGIT POSITIVE INTEGER? 02
OOPS! ENTER AN INTEGER, 10 TO 99.

TWO-DIGIT POSITIVE INTEGER? and so on.

Accept *only* integers in the range 10 to 99, inclusive. Try some of these to test your program.

TWO-DIGIT INTEGER	SUM OF DIGITS	PRODUCT OF DIGITS
10	1	0
28	10	16
99	18	81

PROBLEM #21 SUM AND PRODUCT OF DIGITS OF A THREE-DIGIT NUMBER

If you are a novice programmer, you may wish to first do PROBLEM #20, then this one.

Write a program to compute and print or display the sum of the digits and the product of the digits of a three-digit positive integer. A three-digit positive integer is:

- (1) an integer
- (2) greater than or equal to 100,
- (3) less than or equal to 999.

$$100 \leq X \leq 999$$

Sample RUN.

THREE-DIGIT POSITIVE INTEGER? 123
SUM OF DIGITS = 6
PRODUCT OF DIGITS = 6

THREE-DIGIT POSITIVE INTEGER? 99
OOPS! ENTER AN INTEGER, 100 to 999.

THREE-DIGIT POSITIVE INTEGER? 1234
OOPS! ENTER AN INTEGER, 100 to 999.

THREE-DIGIT POSITIVE INTEGER? 3.14
OOPS! ENTER AN INTEGER, 100 to 999.

THREE-DIGIT POSITIVE INTEGER? 007
OOPS! ENTER AN INTEGER, 100 to 999.

THREE-DIGIT POSITIVE INTEGER? and so on

Accept *only* integers in the range 100 to 999, inclusive. Try some of these to test your program.

THREE-DIGIT INTEGER	SUM OF DIGITS	PRODUCT OF DIGITS
100	1	0
394	16	108
999	27	729

PROBLEM #22 TWO-DIGIT NIVEN NUMBERS

Niven numbers? Until recently, we had never heard of them. Then, the Winter 1980 issue of the MATYC JOURNAL* arrived. The cover story was "Mathematical Discovery and Niven Numbers" by Robert E. Kennedy, Terry A. Goodman and Clarence H. Best.

A Niven number is a positive integer which is divisible by the sum of its digits.

- 24 is a Niven number because $2 + 4 = 6$ is a divisor of 24.
- 37 is *not* a Niven number because $3 + 7 = 10$ is *not* a divisor of 37.

Write a program to compute and print or display all two-digit Niven numbers. We want all Niven Numbers in the range 10 to 99, inclusive.

For more info on Niven numbers, contact Robert E. Kennedy, Mathematics Dept., Central Missouri State University, Warrenburg, MO 64093.

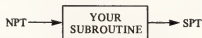
*THE MATYC JOURNAL is published three times a year by the Department of Mathematics/Statistics/Computer processing, Nassau Community College, Garden City, NY 11530. One year (3 issues) for \$8.50.

PROBLEM #23 SUM OF DIGITS SUBROUTINE

Write a subroutine to compute the sum of the digits of a positive integer.

NPT = a positive integer.
SPT = sum of the digits of NPT.

If NPT = 235, then SPT = $2 + 3 + 5 = 10$



Use this program, or a similar program, to check out your subroutine.

```

100 REM *** SUM OF DIGITS OF A POSITIVE INTEGER
200 REM *** ASK FOR AN INTEGER
210 CLS
220 PRINT: INPUT "ENTER AN INTEGER, 1 TO 999999": N
230 IF N <> INT(N) OR N < 0 OR N > 999999 THEN 220

300 REM *** USE SUM OF DIGITS SUBROUTINE
310 NPT = N
320 GOSUB _____ line number of your subroutine.

400 REM *** PRINT SUM OF DIGITS
410 PRINT "THE SUM OF DIGITS OF " NPT "IS" SPT
420 GOTO 220
  
```

Your SUBROUTINE

In the above program, we limit numbers to integers in the range 1 to 999999 (line 230). Change this limit if you wish.

In the January/February, March/April, and May/June issues of *Recreational Computing*, the "Programming Problems and Solutions" series by Bob Albrecht, Don Albers, and Jim Conlan should have included the notation "Copyright 1980 by Bob Albrecht, Don Albers, and Jim Conlan" as they retain personal copyright to the material.

SOLUTIONS TO PROBLEM #1

BY BOB ALBRECHT, DON ALBERS AND JIM CONLAN

The "Programming Problems" section of *RC* began in the Feb-Mar 1980 issue with the following problem.

PROBLEM #1 POSITIVE, NEGATIVE OR ZERO

An easy problem. All we want is a program that asks for a number, then tells you something about the number.

- If you enter a positive number, the computer tells you: YOUR NUMBER IS POSITIVE
- If you enter a negative number, the computer tells you: YOUR NUMBER IS NEGATIVE
- If you enter the number zero (0), the computer tells you: YOUR NUMBER IS ZERO

A solution to this problem is shown below, in TRS-80 BASIC, a version of MicrosoftTMBASIC. With minor editing it will run on the APPLE, ATARI, PET and several other computers.

```

100 REM***PROBLEM #1 POSITIVE, NEGATIVE OR ZERO
110 REM***RECREATIONAL COMPUTING, JAN/FEB 1980
120 CLS

300 REM***ASK FOR A NUMBER, X
310 PRINT: INPUT "NUMBER, PLEASE": X

500 REM***TELL WHETHER NUMBER IS POSITIVE,
    NEGATIVE OR ZERO
510 IF X > 0 THEN PRINT "YOUR NUMBER IS POSITIVE"
520 IF X < 0 THEN PRINT "YOUR NUMBER IS NEGATIVE"
530 IF X = 0 THEN PRINT "YOUR NUMBER IS ZERO"

700 REM***GOTO 'ASK FOR A NUMBER, X'
710 GOTO 310
  
```

Ha! We have pre-empted the solution which immediately leaped to your mind. In fact, you *can't* use the IF statement at all! Your Program must "behave" exactly as ours, but without any IF statements. Let's call the above program Solution #1. Here are six more solutions. In each solution, we will show only the *changes* to be made to Solution #1.

Solution #2 uses the On ... GOTO statement and the SGN function to replace the IF statements in lines 510, 520, and 530.

```

510 ON SGN(X) + 2 GOTO 520, 530, 540
520 PRINT "YOUR NUMBER IS NEGATIVE": GOTO 310
530 PRINT "YOUR NUMBER IS ZERO": GOTO 310
540 PRINT "YOUR NUMBER IS POSITIVE": GOTO 310
  
```

You may also delete lines 700 and 710 of Solution #1.

In Solution #3, we use ON ... GOSUB instead of ON ... GOTO.

```

510 ON SGN(X) + 2 GOSUB 530, 540, 550
520 GOTO 310
530 PRINT "YOUR NUMBER IS NEGATIVE": RETURN
540 PRINT "YOUR NUMBER IS ZERO": RETURN
550 PRINT "YOUR NUMBER IS POSITIVE": RETURN
  
```

Again, delete lines 700 and 710 of Solution #1.

Solutions 1 through 3 are the types of solutions we have become accustomed to seeing for this problem. We rarely see the solutions which follow.

Solution #4 uses a string array, A\$().

```
130 A$(1) = "NEGATIVE"
140 A$(2) = "ZERO"
150 A$(3) = "POSITIVE"

510 N = SIGN(X) + 2
520 PRINT "YOUR NUMBER IS" A$(N)
```

Or, we can pack information into a single string and pick off the appropriate substring, as in Solution #5, below.

```
130 AS = "NEGATIVEZERO POSITIVE"
510 N = SIGN(X) + 1
520 PRINT "YOUR NUMBER IS " MID$(AS, 8*N+1, 8)
```

In some BASICs, of course, you will use the SEG\$ function where we have used MID\$. Or, if you have an ATARI 400 or ATARI 800, replace line 520 with the following.

```
520 PRINT "YOUR NUMBER IS" ; AS(8*N+1, 8*N+8)
```

Next, here is Solution #6

```
510 N = SIGN(X) + 2
520 RESTORE
530 FOR K = 1 TO N
540 READ AS
550 NEXT K
560 DATA NEGATIVE, ZERO, POSITIVE
570 PRINT "YOUR NUMBER IS " AS
```

Hmmm...let's combine the ideas of Solutions 5 and 6. Here is Solution #7.

```
130 FOR K = 1 TO 3
140 READ AS(K)
150 NEXT K
160 DATA NEGATIVE, ZERO, POSITIVE

510 N = SIGN(X) + 2
520 PRINT "YOUR NUMBER IS " A$(N)
```

A few versions of BASIC have a GOTO exp statement, where exp may be any BASIC expression. Solution #8 uses a GOTO exp statement along with clever use of two logical operations.

```
510 "GOTO 530 + 10*(X>0) - 10*(X<0)"
520 PRINT "YOUR NUMBER IS NEGATIVE"
525 GOTO 510
530 PRINT "YOUR NUMBER IS ZERO"
535 GOTO 510
540 PRINT "YOUR NUMBER IS POSITIVE"
545 GOTO 510
```

It works like this:

VALUE OF X	VALUE OF (X>0)	VALUE OF (X<0)
Negative	0	1
Zero	0	0
Positive	1	0

So, you can now compute the corresponding values of $530 + 10*(x>0) - 10*(x<0)$ in line 510.

Some people used logical operations in a different way, illustrated by Solution #9, below:

```
130 AS = "POSITIVE ZERO NEGATIVE"
510 N = 1 + 10*(X>0) - 10*(X<0)
520 PRINT "YOUR NUMBER IS " ; AS(N, N+10)
```

Watch out! Line 510 assumes that a logical expression such as $x>0$ or $x<0$ has the value -1 when true, or 0 when false. Yes, some computers use 1 for true; others use -1 for true. Solution #9 assumes values for $x>0$ and $x<0$, as follows:

VALUE OF X	VALUE OF (X>0)	VALUE OF (X<0)
Negative	0	0
Zero	0	-1
Positive	-1	0

Is your computer a 1 or a -1?

Oh yes, almost forgot. In line 520 you may have to change A(N,N+10)$ to MID(A$, N, 10)$.

Well, our ingenious readers sent us some real surprises. Solution #10 is Richard Gilman's "off-the-wall" approach. He used ON-ERROR to achieve a GOTO. The appropriate errors are generated by $A = SQR(x)$ and $A = 1/x$.

```
10 REM *** SOLUTION C TO PROBLEM #1
20 CLS
30 REM *** ASK FOR A NUMBER, X
50 INPUT "NUMBER, PLEASE "; X
100 REM *** CHECK FOR NEGATIVITY. LINE 120 CAUSES ERROR IF X
    IS NEGATIVE.
110 ON ERROR GOTO 500
120 A = SQR(X)
200 REM *** CHECK FOR ZERO. LINE 220 CAUSES ERROR IF X IS ZERO
210 ON ERROR GOTO 400
220 A = 1/X
300 REM *** IF NUMBER IS NEITHER NEGATIVE OR ZERO IT MUST BE POSITIVE.
310 PRINT "YOUR NUMBER IS POSITIVE"
320 GOTO 710
500 REM *** ERROR SUBROUTINE CAUSED BY NEGATIVE NUMBER IN LINE 120.
510 PRINT "YOUR NUMBER IS NEGATIVE"
520 RESUME 710
600 REM *** ERROR SUBROUTINE CAUSED BY ZERO IN LINE 220.
610 PRINT "YOUR NUMBER IS ZERO"
620 RESUME 710
700 REM *** GOTO 'ASK FOR A NUMBER, X'
710 GOTO 50
999 END
```

Solution #11. Bob Baker and students Sarah Barter and Debbie Hauth used only string functions and arithmetic to isolate the minus sign and compute an index for the string to be printed.

```
100 REM***PROBLEM #1 POSITIVE, NEGATIVE OR ZERO
110 REM***RECREATIONAL COMPUTING, JAN/FEB 1980
120 CLS
300 REM***ASK FOR NUMBER, X
310 PRINT: INPUT "NUMBER, PLEASE"; X
500 REM***TELL WHETHER NUMBER IS POSITIVE, NEGATIVE OR ZERO
510 XS = STR$(X) : REM CONVERT X TO A STRING
520 XS = LEFT$(XS,1)
530 XS = XS + "!" : REM MAKE X A POSITIVE OR NEGATIVE 1
540 XS = VAL(XS) : REM Y BECOMES + OR - DEPENDING ON X
550 Y = X*(XS) : REM CHANGE THE SIGN OF X
560 XS = STR$(X) : REM CONVERT -X TO A STRING
570 XS = LEFT$(XS,1) : REM ISOLATE THE SIGN OF -X
580 XS = XS + "!" : REM MAKE -X A POSITIVE OR NEGATIVE 1
590 Z = VAL(XS) : REM 2 IS + OR - DEPENDING ON -X
600 X = (Z*(XS))/2 + 1 : REM THIS EQUATION CAUSES THE ORIGINAL X
    TO BE 1, 2 OR 3. ZERO AND NEGATIVE ZERO
    MAKE Y AND Z BOTH EQUAL TO POSITIVE 1.
    THEREFORE Y - Z BECOMES 0 FOR ZERO,
    2 FOR A POSITIVE NUMBER, AND -2 FOR A
    NEGATIVE NUMBER
601
610 XS(3) = "YOUR NUMBER IS POSITIVE"
620 XS(2) = "YOUR NUMBER IS ZERO"
630 XS(1) = "YOUR NUMBER IS NEGATIVE"
640 PRINT XS(X)
700 REM***GOTO 'ASK FOR NUMBER, X'
710 GOTO 310
999 END
```

Solution #12. Perhaps the most exotic solution was contributed by John J. Wavrik. Since his TRS-80 (level I) doesn't have the computed GOTO X command, he wrote a GOTO with a dummy line number: GOTO 100. Then, he peeked around till he found where the digits of the dummy line number were stored and poked in a new, computed value. Here are his comments and the program.

"One of the handy features I sometimes would like in BASIC but which isn't found in my TRS-80 is a GOTO X...a 'compute GOTO' The effect of such an instruction can be

obtained by a program which actually changes itself. Here is an example of such a program for a Level II TRS-80:

```
10 INPUT N
20 S = SGN(N) + 49
30 POKE 17173,S
40 GOTO 100
100 PRINT "NEGATIVE";GOTO 10
101 PRINT "ZERO!";GOTO 10
102 PRINT "POSITIVE";GOTO 10
```

The number 17173 is the location in memory of the last '0' in line 40. When you run the program, the POKE statement on line 30 will change the GOTO to read GOTO 100 if the number you put in is negative, GOTO 101 if the number you put in is zero, GOTO 102 if the number you put in is positive."

To use this idea in your version of BASIC, you must write a program with a "dummy value" for the address on line 30. Then you must find out where in memory the BASIC text is stored and, within that block, where, in particular, the GOTO 100 command on line 40 is. I used FOR I=17129 TO 17200: PRINT I;PEEK(I);NEXT to get a list of memory addresses and their contents. The TRS-80 stores the BASIC text starting at 17129 for a Level II machine. The text consists of a 2 byte number showing the address of the start of the next line, a 2 byte number representing the line number, a collection of bytes for the instructions and, finally, a 0 to indicate the end of the line. All of the BASIC commands are relayed by 1 byte "tokens." 141 is the token for GOTO. Numbers (like 100) are just represented by the ASCII codes for the digits (so 49 48 48 is the representation for 100). Once the location of the final 0 is determined, the dummy address in the POKE statement is replaced by the correct address."

Solution #13. Raymond Schuman's two line solution was the shortest submitted. His program uses SGN(X) to compute the ASCII number for the appropriate response character '+', '0' or '-'. Exceedingly ingenious!

```
10 INPUT X
20 PRINT CHR$( - 4*SGN(X) - SGN(X) + 48)
```

Here are our problem-solvers for **PROBLEM #1** showing who used which method or methods.

	1	2-3	4-5	6	7	8	9	10	11	12	13
Baker, et al.										X	
Bishop			X								
Canova								X			
Daley						X					
Day											
Gilman		X							X		
Hawkinson			X								
Johnson								X			
Langford				X							
Lemay									X		
Lindsay		X	X			X					
Perkins						X					
Schuman											X
Schwanbeck								X			
Wavrik		X	X							X	
The Editors	X	X	X	X	X						

What To Do After You Hit Return

PCC's First Book of Computer Games

Fifth Printing

What To Do After You Hit Return is a compendium of games for kids and adults— games to develop decision making and problem solving skills for new and unexpected situations.

Includes complete listings in BASIC. 184pp. 5th printing. Retail price: \$10.95. (Add \$1.25 book rate or \$2.50 UPS for shipping. California residents add 6% sales tax.)

Please send me _____ copy(ies) of PCC's *What To Do After You Hit Return* for which I enclose \$10.95 plus shipping charge for each copy ordered.

Send your order to People's Computer Company, Dept. J7, Box E, Menlo Park, CA 94025.

Name _____
 Address _____
 City/State _____ Zip _____
 Or Charge it! ☐ Visa ☐ MasterCard ☐ American Express
 No. _____ Expires _____
 Signature _____

The Further Adventures
of

FORTMAN

Volume III

Episode 12

By L. SCHNEIDER & T. VOROS

In our last episode, Fortman Man had at last penetrated the innermost chambers of the Glitchmaster's stronghold—the Master Control room of Tesla Tower—to be confronted, at last, by the Glitchmaster himself!

And yet... even F-Man must PAUSE in surprise... for the greatest power of a Glitch lies in being able to appear as who was never expected... and certainly the ridiculously harmless-looking creature encountered by Our Hero was *not* what he was expecting!

Nor was he by any means harmless! For no sooner had F-Man stated his program objective of terminating the Glitchmaster's control of MicroprocessorLand than the latter began to produce an amazing array of weapons—each coming literally out of nowhere, and each allowing the Glitchmaster to launch yet another deadly attack against Our Hero!

Yet in the face of an Off-By-One Inducer, a Pointless Pointer, and a Noise Pulse Generator, Fortman Man remains untermiated! And yet, the Glitchmaster only announces calmly that he has only just begun... and they may now proceed with the real battle!

Let's see now... I wonder how you'd fare against my Data Destroyer, F-Man?

And proceed he does... with an amazing series of strange and deadly weapons, one after another... and each of them aimed quite accurately at Our Hero!

... or maybe a Dimension Distorter will do the trick

... and here's an oldie-but-goodie: my Function Freezer!

I'll bet *this*'ll slow down your throughput rate a bit or two!

ZORCH!

Whew! Catching this Glitch isn't going to be as easy as I thought!

Fortunately my redundant code enables me to survive such recursive attacks!

And then, with unexpected suddenness, the assaults end! And although worn down by a bit or two, F-Man again repeats the iteration of his terms...

Well then... are you ready to give up yet?

Put an end to this noisy conflict and come quietly!

puff... puff...

Yet the reply comes just as calmly as before...

Hmmm... an interesting problem! It seems that you, Fortman Man, are impervious to all my best weapons!

I must take some real time and think about this!

Seeing the opportunity in his foe's hesitation, F-Man once again enters RUN mode, incrementing himself at high speed towards his opponent...

That's what you think, Glitchmaster!

Your processing time has expired... now it's time for you to be output for good!

Yet in the face of F-Man's swift attack the Glitchmaster makes no move to defend himself... nor does he make any attempt to relocate himself out of the execution path of the oncoming foe...

This is it, Glitchmaster! You have executed all your tricks; and now it's my turn to...

Ah, but there you are incorrect, F-Man. You seem to forget that I always have the final, last resort of a Glitch! You see...

...I can...

Wha... where'd he go? I was certain that...

Of course! I should have remembered! A glitch always disappears just as you get close enough to fix it for good!

Perhaps my memories were not as fully restored as I had thought!

But the space is not empty for long, as within nanoseconds there comes a CALL from the corridors outside... and F-Man needs little time to identify its source code...

Fortran Man! Where are you?

In here, Billy!

And abruptly, F-Man finds himself standing in an empty program space...

Quickly F-Man explains...

...and so you see, the Glitchmaster could not defeat me by force... so he fled!

Had I only remembered in time his ability to escape in that way, I may have caught him yet!

Yes, F-Man, trying to catch a Glitch can be very aggravating... but it is of no matter! We shall destroy the devices here and disconnect Tesla Tower... so it will be impossible for him to ever RETURN to our land!

And as the General departs to regroup his Resistance decades and take on the job of altering the current status of Capital City to handle the recently liberated Lower Cases, it is Linpa's turn to speak out...

Yes, F-Man, the threat is gone... and the Land of the Little People is once again noise-free... thank to you!

And now that that's done with, Billy and I have some catching up on old times to do... don't we, Billy?

oh? er... what?

Another job terminated successfully!

Now I can RETURN at last to my home in 360 City, and relax for a while before carrying on the fight for Truth, Justice and the Algorithmic Way! Only... I wonder... what will come next?

And that, dear readers, only time will tell. For now, though, it is time for us to say:

STOP
END

...disappear!

POOF

Thank heavens we traced you to this location! Is your code intact? What a mess!

Yes... and where is the Glitchmaster???

And soon thereafter he is linked to by not only Billy Basic, but Linea and General Wirewound as well!

So once again Fortran Man is left alone in the now-deserted program space, responding to the falling edge of the clock as it descends behind the distant Monolithic Mountains, and Peace settles into every gate and latch element in the land...

F



ALSO PUBLISHED BY PEOPLE'S COMPUTER COMPANY:

DR. DOBB'S JOURNAL of COMPUTER Calisthenics & Orthodontia

If you're tired of plugging canned programs into your computer, *Dr. Dobb's Journal* is the magazine for you! We publish software that will allow you to get "inside" your machine. You will be able to make it do anything you want it to do. *Dr. Dobb's* is a serious magazine for those people who really want to learn as much as they can about their microcomputer.

Recent issues have included: • An All-CP/M Issue, including an overview of CP/M-compatible software, the history of CP/M, utilities, interpreters, etc. • Mathematical Typography by Donald Knuth • A Microprocessor-Assisted Graphics System • Interfacing a Home Computer to a Large Scale System • Improvements to 8080 PILOT

OUR READERS SAY:

"I received my introduction to your remarkable journal and was as pleased as a Lovecraft wizard coming upon a copy of the *Necronomicon*. Really, yours is the only real computer journal going. *Dr. Dobb's* is to the other "small systems" periodicals as *Scientific American* is to *Popular Mechanics*."

"*Dr. Dobb's Journal* is a truly impressive periodical unlike others I've read. It contains valuable and useful software. I look forward to its delivery every month."

See insert card inside this issue.



Recreational Computing
Box E
Menlo Park, CA 94025

SECOND CLASS MAIL
MAGAZINE

08607DITTECO8

30D2

3053

CHARLES S DITTELL

SE81

BOX 1448

TRENTON, NJ 08607

Address Correction Requested